DESIGN AND IMPLEMENTATION OF WEB BASED PHISHING DETECTION SYSTEM USING RULE BASED AND LLM APPROACHES



Conduct by:

Name : MUHAMMAD ASWIN JALU

Student ID : 21523009

INFORMATICS STUDY PROGRAM - INTERNATIONAL BACHELOR PROGRAM

FACULTY OF INDUSTRIAL TECHNOLOGY

ISLAMIC UNIVERSITY OF INDONESIA

2026

SUPERVISOR ENDORSEMENT PAGE

DESIGN AND IMPLEMENTATION OF WEB-BASED PHISHING DETECTION SYSTEM USING RULE-BASED AND LLM APPROACHES

THESIS

Conduct By:

Name	: MUHAMMAD ASWIN JALU
Student ID	: 21523009

Yogyakarta, August 3rd, 2026

Advisor,

(Erika Ramadhani, S.T., M.Eng.)

EXAMINER ENDORSEMENT PAGE

DESIGN AND IMPLEMENTATION OF WEB-BASED PHISHING DETECTION SYSTEM USING RULE-BASED AND LLM APPROACHES

THESIS

Has been defended in front of the examiners as one of the requirements to obtain a Bachelor of Informatics degree from the Undergraduate Program in Informatics at the Faculty of Industrial Technology, Universitas Islam Indonesia

Yogyakarta, August 3rd, 2026

Chair

Erika Ramadhani, S.T., M.Eng. _____

Examiner 1

Feri Wijayanto, Dr., S.T., M.T. _____

Examiner 2

Taufiq Hidayat, S.T., M.C.S., Ph.D. _____

Acknowledged by,

Head of Undergraduate Program in Informatics

Faculty of Industrial Technology

Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

AUTHENTICITY STATEMENT

The undersigned:

Name	:	MUHAMMAD ASWIN JALU
Student ID	:	21523009

Final project with title:

DESIGN AND IMPLEMENTATION OF WEB-BASED PHISHING DETECTION SYSTEM USING RULE-BASED AND LLM APPROACHES

Stating that all components and contents in this final project are my own work. If in the future it is proven that some parts of this work are not my own work, the final project submitted as my own work is ready to be withdrawn and ready to bear any risks and consequences.

Thus this statement letter is made, hopefully it can be used properly.

Yogyakarta, August 3rd, 2026



(MUHAMMAD ASWIN JALU)

DEDICATION

Bismillahirrahmanirrahim

Alhamdulillāhi Rabbil 'Ālamīn, I offer all praise and gratitude to Allah SWT for His abundance of mercy, taufik, guidance and inayah so that I can be given health, strength, patience and ease in completing this Final Assignment. Shalawat and greetings are always poured out to the Prophet Muhammad SAW, along with his family, friends and all his people until the end of time.With great gratitude and humility, I dedicate this simple work to my beloved Father and Mother, who have always been a source of love, strength, inspiration and enthusiasm in every step of my life.

Thank you for every prayer that never stops being said, every sacrifice that is given without getting tired, as well as sincere love and endless support throughout your educational journey until completing this Final Assignment. All the achievements I have achieved are the result of the struggle, sincerity and pleasure of my parents who always accompany every step of the writer.I hope this final project serves as a form of devotion and an expression of gratitude. While this work will never be able to repay all the sacrifices, love, and affection they have given me, I hope this achievement can be the beginning of bringing happiness and pride to my parents.

May Allah SWT always bestow His mercy, health, blessings, happiness, and protection upon my parents, repay every kindness and sacrifice with abundant rewards, and keep our family always under His protection and blessings. With love, respect, and gratitude, this work is dedicated to my beloved parents.

MOTTO

Strive to the best of your ability, because the results of honest effort will always be more meaningful than success achieved quickly. Every small step taken earnestly will bring us closer to our goal. No process is truly wasted as long as we continue to learn, improve ourselves, and never give up when faced with difficulties. Success is not just about how quickly someone reaches the finish line, but about how they are able to persevere, fight, and overcome every challenge with integrity. Therefore, don't be afraid to start from scratch, don't be ashamed to learn from mistakes, and never give up just because the journey feels difficult. Believe that every effort made with good intentions, hard work, discipline, and prayer will yield the best results in time. Take failure as a lesson, use challenges as motivation, and view every process as part of the journey to becoming a better person.

لَهُۥ مُعَقِّبَٰتٌ مِّنۢ بَيْنِ يَدَيْهِ وَمِنْ خَلْفِهِۦ يَحْفَظُونَهُۥ مِنْ أَمْرِ ٱللَّهِ ۗ إِنَّ ٱللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّىٰ يُغَيِّرُوا۟ مَا بِأَنفُسِهِمْ ۗ وَإِذَآ أَرَادَ ٱللَّهُ بِقَوْمٍ سُوٓءًا فَلَا مَرَدَّ لَهُۥ ۚ وَمَا لَهُم مِّن دُونِهِۦ مِن وَالٍ ١١

For each one there are successive angels before and behind, protecting them by Allah’s command. Indeed, Allah would never change a people’s state ˹of favour˺ until they change their own state ˹of faith˺. And if it is Allah’s Will to torment a people, it can never be averted, nor can they find a protector other than Him.

“Indeed, Allah will not change the condition of a people until they change what is in themselves.” (QS. Ar-Ra'd: 11).

فَإِذَا فَرَغْتَ فَٱنصَبْ ۝٧

Strive with your own abilities, because effort from the heart is a source of pride, and the results obtained through hard work will always be the most valuable achievement.When you are finished (with one virtue), keep working hard (for another virtue)

“So when you have finished (from one business), keep working hard (on other matters).” (QS. Al-Insyirah: 7)

"Try to the best of your ability, because effort that comes from the heart is pride, and the results obtained through hard work will always be the most valuable achievement."

FOREWORD

All praise and gratitude are due to Allah SWT for His blessings, guidance, and mercy, which have enabled the author to complete this final project entitled "Design and Implementation of a Web-Based Phishing Detection System Using a Rule-Based Approach and LLM." This report has been prepared as one of the requirements for a Bachelor's degree in Informatics at the Islamic University of Indonesia.

The primary objective of this research is to design, implement, and evaluate a web-based phishing detection system that combines rule-based analysis with Large Language Model (LLM) technology. The proposed hybrid approach aims to improve phishing URL detection by integrating traditional rule-based methods with the contextual reasoning capabilities of LLM, thereby providing more accurate and informative analysis results to users.

During the completion of this research, I faced several challenges, particularly in integrating various technologies, collecting and validating phishing datasets, optimizing system performance, and evaluating the effectiveness of the hybrid detection model. These challenges required continuous learning, experimentation, and careful problem-solving throughout the research process.

I would like to express my sincere gratitude to all parties who contributed to the completion of this final project. Special appreciation is extended to my supervisors for their valuable guidance, constructive suggestions, and continued encouragement throughout the research process. I also thank the lecturers in the Department of Informatics, Islamic University of Indonesia, for the knowledge and insights they provided throughout my studies. Furthermore, my deepest gratitude goes to my family and friends for their unwavering support, motivation, patience, and prayers.

I acknowledge that this report is not without limitations and may still contain shortcomings. Therefore, we welcome constructive criticism and suggestions for future improvements. It is hoped that this research will contribute to the field of cybersecurity, particularly in phishing detection, and serve as a useful reference for future researchers and practitioners interested in developing intelligent phishing detection systems.

Yogyakarta, August 3rd, 2026



(MUHAMMAD ASWIN JALU)

ABSTRACT

This research is motivated by the increasing threat of cybercrime, particularly phishing attacks that exploit the rapid development of information technology to obtain sensitive user information, such as login credentials and personal data, through fraudulent websites or malicious links. One of the approaches that can be used to detect phishing attacks at an early stage is the analysis of suspicious URL patterns and structures (Mohammad et al., 2014). Therefore, this study aims to design a web-based system capable of identifying phishing site patterns using a hybrid approach that combines rule-based methods and Large Language Models (LLMs). The rule-based method is utilized to extract indicators or digital traces from URL structures, including the use of IP addresses instead of domain names, the presence of the "@" symbol, abnormal URL lengths, sensitive keywords, URL shorteners, excessive numbers of subdomains, and hyphens in domain names (Aljofey et al., 2020; Mohammad et al., 2014). The results of this initial analysis are used as indicators of predictable phishing patterns. However, when the rule-based method cannot adequately identify suspicious indicators, the LLM performs advanced analysis to evaluate unpredictable phishing patterns and generate a more comprehensive final decision. The system is designed to produce two types of analysis results, namely rule-based detection results and LLM-based analysis results. In addition, the system provides data storage functionality and PDF report generation as a form of digital documentation. The results indicate that rule-based methods alone have limitations because they can only detect predefined indicators. Therefore, integrating LLMs improves the system's ability to identify more complex phishing patterns and previously unseen threats that are not explicitly defined within the rule set.

Keywords: Cybersecurity, Large Language Model (LLM), Phishing, Rule-Based, URL Analysis.

GLOSSARY

- Accuracy : The ratio of correctly predicted observations to the total observations within an evaluation dataset, indicating overall classification correctness.
- Backend : The server-side layer of an application responsible for business logic execution, data processing, routing, and database interactions, which remains invisible to the end user.
- Confusion Matrix : A tabular layout used to visualize the performance of a classification model by cross-referencing actual ground-truth labels against the system's predicted results across distinct categories.
- Containerization : An OS-level virtualization method used to package software code together with all its necessary dependencies, libraries, and configuration files into an isolated environment.
- Deterministic : A system property or algorithmic process where the exact same initial input will consistently yield identical, predictable outputs without any random variation.
- Digital Forensics : A branch of forensic science encompassing the recovery, preservation, identification, processing, and analysis of digital evidence stored on electronic devices or captured from network traffic.
- Docker : An open-source software platform used to automate the deployment, scaling, and management of applications inside standardized, lightweight container environments.
- F1-Score : The harmonic mean of precision and recall, serving as a balanced metric to evaluate the performance of a classification model, particularly when dealing with highly imbalanced datasets.
- Frontend : The client-side presentation layer of a software system that renders the user interface within a web browser, allowing direct user interaction.
- Heuristics : Practical, rule-of-thumb criteria or experience-based logical constraints utilized by a system to identify specific traits, indicators, or structural patterns without running heavy computations.
- Hybrid Architecture : A software system design that bridges and integrates two or more distinct technological methodologies—such as coupling deterministic static rules with non-deterministic artificial intelligence—to perform a unified analytical task.

- JSON (JavaScript Object Notation) : A lightweight, text-based, language-independent data-interchange format that is highly structured and easily readable by both humans and automated parsers.
- Large Language Model (LLM) : An advanced deep learning artificial intelligence architecture trained on massive textual datasets, designed to process, interpret, contextualize, and generate human-like natural language.
- MySQL : An open-source relational database management system (RDBMS) that utilizes Structured Query Language (SQL) for storing, organizing, and querying relational data records.
- Phishing : A form of social engineering cyberattack where malicious actors deploy deceptive links and spoofed interfaces to impersonate legitimate entities, tricking targets into disclosing sensitive corporate or personal data.
- Precision : The ratio of correctly predicted positive instances to the total predicted positive instances, measuring the degree of reliability or confidence in a system's specific threat classification.
- Prompt Engineering : The systematic process of structuring, refining, and optimizing textual instructions and parameters provided as inputs to a Large Language Model to guide its contextual reasoning toward a specific, standardized output.
- Recall : The ratio of correctly predicted positive instances to all actual positive instances within a dataset, measuring a system's empirical threat detection capability.
- Rule-Based : A computing methodology that relies exclusively on a predefined, static repository of conditional IF-THEN logic strings to parse inputs and output deterministic assessments.
- Temperature : A hyperparameter used to control the randomness, creativity, or strict determinism of a Large Language Model's text generation, where values closer to 0 enforce highly predictable and factual outcomes.
- Top-p : A sampling hyperparameter (also called nucleus sampling) that bounds the token selection pool of a Large Language Model to a specific cumulative probability threshold, eliminating highly improbable lexical choices.

TABLE OF CONTENTS

TITLE PAGE i
SUPERVISOR ENDORSEMENT PAGE ii
EXAMINER ENDORSEMENT PAGE iii
AUTHENTICITY STATEMENT iv
DEDICATION v
MOTTO vi
FOREWORD vii
ABSTRACT viii
GLOSSARY ix
TABLE OF CONTENTS xi
LIST OF TABEL xiii
LIST OF FIGURES xiv
CHAPTER I INTRODUCTION 1
1.1 Background 1
1.2 Problem Formulation 2
1.3 Research Objectives 3
1.4 Problem Limitations 3
1.5 Research Benefits 3
CHAPTER II LITERATURE REVIEW 4
2.1 Relevant Research 4
2.2 Problem Statement 8
2.3 Theoritical Foundations 8
2.3.1 Phishing 8
2.3.2 Characteristics and Structure of URLs 9
2.3.3 Rule-Based Detection 9
2.3.4 Large Language Model (LLM) 10
2.3.5 Hybrid Decision Making Model 11
CHAPTER III METHODOLOGY 12
3.1 Research Methods 12
3.2 Requirements 12
3.2.1 Hardware Requirements Analysis 12
3.2.2 Software Needs Analysis 13
3.2.3 System Needs Analysis 13
3.3 System Designs 13
3.3.1 System Architecture 14
3.4 Workflow of the Work System 23
3.5 Rule-Based System 24
3.5.1 Categories Rule-Based 25
3.5.2 LLM Analysis 29
3.5.3 Final Decision 29
3.5.4 Application Design 30
3.6 Implementation 32
3.7 Functionality Testing 33
3.8 Dataset Preparation 37
3.9 Peformance Evaluation 43
CHAPTER IV DISCUSSION AND RESULTS 46
4.1 How The System works 46

4.2 System Implementation 51
4.3 System Test Results 51
4.3.1 Input Test URL 52
4.3.2 Rule-Based Analysis 54
4.3.3 LLM Analysis 55
4.3.4 Hybrid (Auto-Bypass) 56
4.3.5 Hybrid (Full Validation) 57
4.3.6 Database Storage 58
4.3.7 Analysis History 59
4.3.8 Export PDF 60
4.3.9 Docker Backend 61
4.3.10 Results Display 62
4.4 System Effectiveness 63
4.4.1 Confusion Matrix Distribution of Test Data Hybrid (Auto-Bypass) 64
4.4.2 Evaluation Metric Value Analysis 66
4.4.3 Methode Performance Comparison 67
4.4.4 Label Normalization and Risk Mapping 68
4.4.5 Analysis of Classification Errors 69
4.5 Conclusion 71
CHAPTER V CONCULSION 73
5.1 Conslusion 73
5.2 Suggestions 74
REFERENCE 75
APPENDIX 77

LIST OF TABEL

Tabel 3.1 GPT-4o-mini Promt Implementation 17
Tabel 3. 2 Risk Classification Function 21
Tabel 3.3 GPT-4o-mini Configuration Parameters 21
Tabel 3.4 List Rule-Based 25
Tabel 3. 5 List of risk level classifications based on Rule-Based scores 27
Tabel 3. 6 Characteristics and weights of phishing detection 28
Tabel 3.7 Feature Testing Secenario 33
Tabel 3.8 Functionality Testing 36
Tabel 3.9 URL Dataset for System Evaluation 38
Tabel 3.10 Label Normalization from Original Dataset to System Classification 42
Tabel 3.11 System Prediction Normalization for Performance Evaluation 43
Table 4.2 Evaluation Metric Value Analysis 66
Table 4.3 Method Performance Comparison 67
Table 4.4 Table Label Normalization and Risk Mapping 68

LIST OF FIGURES

Figure 3.1System Architecture 14
Figure 3. 2 Prompt Engineering Workflow in Hybrid Method 22
Figure 3.3 Workflow of the Work System 23
Figure 3. 4 Rule-Based System 24
Figure 3.5 architecture of the hybrid model-based phishing detection system 29
Figure 3.6 Phising Detection System 31
Figure 3. 7 MySQL 9.7.1 32
Figure 4.1 Home Page 46
Figure 4.2 Analysis Mode Selection 47
Figure 4.3 URL Analysis Process and Results 48
Figure 4.4 Storage of Analysis History Results 49
Figure 4.5 PDF Report Genertor 50
Figure 4.6 Analysis Result 52
Figure 4.7 URL Validation 53
Figure 4.8 Rule-Based Analysis 54
Figure 4. 9 LLM Analysis 55
Figure 4.10 Hybrid (Auto-Bypass) 56
Figure 4.11 Hybrid (Full Validation) 57
Figure 4.12 Data Storage 58
Figure 4.13 Analysis History 59
Figure 4.14 Export PDF 60
Figure 4.15 Docker Backend 61
Figure 4.16 Results Display 62
Figure 4.17 Confusion Matrix 64

CHAPTER I
INTRODUCTION

1.1 Background

The rapid development of information technology and the use of the internet have provided various conveniences in people's activities, such as communication, online transactions, and access to information quickly and widely. However, this development is also accompanied by an increasing threat of cybercrime, one of which is phishing. Phishing is a fraud technique that uses fake websites, malicious links, or information manipulation to obtain sensitive user data, such as usernames, passwords, banking data, and other personal information (Mohammad et al., 2014). Based on various studies, phishing attacks continue to increase and become one of the most frequently used cybersecurity threats because they are able to target various groups of internet users (Elder, 2009).

Several previous studies have developed phishing detection systems using a variety of approaches, such as machine learning, deep learning, and rule-based methods.(Noori et al., 2023) showed that the Random Forest algorithm is able to produce high accuracy in detecting phishing sites compared to other classification methods. Another study by (Noori et al., 2023) also shows that the Random Forest-based approach is effectively used in the classification of malicious URLs on web-based systems. In addition, (Aljofey et al., 2020) explain that URL structure analysis can be used as an early indicator in detecting phishing, such as the use of IP addresses instead of domains, the "@" symbol, unnatural URL lengths, the use of sensitive keywords, the excessive number of subdomains, and the use of hyphens on domains. The rule-based approach is considered to be able to provide a fast, lightweight, and effective detection process to detect predictable phishing patterns.

However, rule-based methods have limitations because they are only able to detect phishing indicators that have been specified in the system rules. Modern phishing attacks are constantly evolving with a variety of new manipulation techniques that don't always fit into the patterns or indicators that are already available. This condition causes rule-based methods to be less flexible in detecting new phishing patterns or unpredictable patterns. (Aljofey et al., 2020) explain that rule-based methods are very effective for detecting known phishing patterns, but still have limitations in recognizing new patterns that have not yet been defined in system rules.

Based on these problems, this study proposes to design a web-based phishing site identification system using a hybrid approach between the rule-based method and the Large Language Model (LLM). Rule-based methods are used to extract indicators or digital traces from URL structures, such as detecting the use of IP addresses, the "@" symbol, URL length, sensitive keywords, URL shorteners, number of subdomains, and hyphens on domains. The results of the initial analysis are then used as predictable phishing indicators. Furthermore, Large Language Model (LLM) is used to conduct advanced analysis of unpredictable phishing patterns by understanding URL patterns more thoroughly and contextually. This approach is expected to improve the system's ability to detect phishing more adaptively than using a single method.

Through the combination of these two methods, the system is expected to be able to produce a phishing identification process that is more effective, adaptive, and interpretive than using a single method. In addition, the system is also designed to produce analysis reports in PDF format as digital documentation, so that it can help users understand the results of phishing detection more clearly and systematically. Based on the results of the tests conducted, the integration of rule-based methods and LLMs is able to improve the system's ability to detect phishing patterns, both known and new patterns that have not been defined in the system rules.

1.2 Problem Formulation

The purpose of this study is to design and develop a web-based phishing identification system that is able to classify websites as phishing or non-phishing using a combination of rule-based methods and Large Language Model (LLM). This research also aims to develop a system that is able to identify and analyze phishing patterns based on the URL structure, the appearance and content of the site, as well as the pattern of text or language used on the website. In addition, this study aims to produce more comprehensive phishing analysis and classification results, where rule-based methods are used to detect predictable phishing indicators, while LLM is used to analyze unpredictable phishing patterns so that it is able to produce a more systematic and clear final decision.

1.3 Research Objectives

The purpose of this study is to design and develop a web-based phishing identification system that is able to classify websites as phishing or non-phishing using a combination of rule-based methods and Large Language Model (LLM). This research also aims to develop a system that is able to identify and analyze phishing patterns based on the URL structure, the appearance and content of the site, as well as the pattern of text or language used on the website. In addition, this study aims to produce more comprehensive phishing analysis and classification results, where rule-based methods are used to detect predictable phishing indicators, while LLM is used to analyze unpredictable phishing patterns so that it is able to produce a more systematic and clear final decision.

1.4 Problem Limitations

In order for the research to be more targeted and not too broad, this study has several limitations. The system is designed only to generate a classification of sites as phishing or non-phishing based on the analysis of the URL structure and certain predefined indicators. The phishing analysis in this study was limited to simple URL patterns, site appearance and content, as well as text patterns or language used, without conducting an in-depth analysis of network traffic or server security. In addition, the methods used are only limited to rule-based approaches and Large Language Model (LLM) without making comparisons with other machine learning or deep learning methods . The LLM model used in this study is GPT-4o-mini as an advanced analysis module against unpredictable phishing patterns.

1.5 Research Benefits

This research is expected to provide benefits in helping users recognize and understand indications of phishing on websites through the analysis of URL structures and other digital indicators. The developed system can be used as a supporting medium in providing information on website classification in the form of phishing or non-phishing as a form of early warning against phishing threats. In addition, this research is also expected to help users understand phishing patterns that are commonly used on websites and provide more understandable analysis results through a combination of rule-based methods and Large Language Model (LLM). From the academic side, this research is expected to be a reference in the development of a phishing detection system based on a hybrid approach between rule-based methods and LLM technology in the field of cybersecurity.

CHAPTER II
LITERATURE REVIEW

2.1 Relevant Research

Research on phishing detection has been conducted using various approaches, ranging from rule-based methods, machine learning, to deep learning. These studies provide an important foundation for the development of the phishing detection system in this research. Most previous studies have focused on URL structure analysis as a primary indicator for identifying phishing websites because URLs can provide early information about website characteristics without requiring resource-heavy content analysis.

Harahap et al. (2024) developed a web-based phishing link detection system using the Random Forest algorithm. The results showed that the method was capable of handling large amounts of data and effectively utilizing URL features in the phishing classification process. Similarly, Windarni et al. (2023) compared the Naïve Bayes, Decision Tree, and Random Forest algorithms for phishing website detection. Their findings demonstrated that Random Forest achieved the highest accuracy rate of 96.3%, indicating its effectiveness in malicious URL classification. In addition, Adiputra and Setiawan (2023) proposed an Improved Random Forest approach for web-based malicious URL classification, which improved classification performance compared to the standard Random Forest method. However, their approach did not incorporate contextual or semantic analysis during the detection process.

Further research by Noori et al. (2023) explored advanced supervised machine learning models, reinforcing the potential of ensemble techniques in phishing URL classification while identifying vulnerabilities to advanced text obfuscation. Likewise, Lukito and Handaya (2022) compared XGBoost and Random Forest algorithms and found that XGBoost with hyperparameter tuning achieved a high accuracy of 96%.

Natural Language Processing (NLP)-based approaches have also been applied to phishing detection to analyze textual patterns. Aryanti and Nabila (2023) utilized the Support Vector Machine (SVM) algorithm combined with the TF-IDF technique to classify phishing URLs. Using a dataset of 11,430 URLs, the study demonstrated that textual URL representation can significantly improve phishing detection effectiveness. In addition, Aljofey et al. (2020) proposed a Character-Level Convolutional Neural Network (CNN) model capable of

automatically extracting URL features without manual feature engineering, thereby improving both efficiency and detection accuracy.

Several studies have focused specifically on lexical URL analysis. Khonji and Iraqi (2011) conducted lexical URL analysis through URL component tokenization to distinguish phishing websites from legitimate ones. Their results indicated that URL structural characteristics could be used as effective early indicators for phishing detection. Similarly, Ma et al. (2011) showed that statistical and lexical URL features can successfully detect phishing websites without relying on static blacklist databases, offering high accuracy through automated statistical profiling.

In addition to machine learning and deep learning approaches, rule-based methods have been widely utilized for deterministic verification. Mohammad et al. (2014) identified crucial phishing indicators such as the use of IP addresses, excessive URL length, and the presence of the "@" symbol. The study introduced a heuristic-based weighting approach that later became the structural foundation for rule-based phishing detection methods. Furthermore, previous research by Pradana et al. (2019) on Rule-Based Attribute Checking demonstrated that rule-based systems are highly effective for identifying known, static phishing indicators such as suspicious URL structures and IP-based URLs. However, these deterministic methods remain limited when trying to detect previously unseen or zero-day phishing patterns.

Other researchers have approached phishing detection from an infrastructure perspective. Garera et al. (2007) developed a phishing detection framework that combines infrastructure features and webpage relationship analysis to identify phishing activity. The study showed that integrating these features can improve detection performance, although it relies heavily on real-time third-party network lookups. Moore and Clayton (2011) examined the impact of publicly available information on phishing attacks and defense mechanisms. Their research focused on phishing ecosystems and attack behaviors at the internet infrastructure level but did not provide a direct automated detection mechanism.

Furthermore, Salvi et al. (2021) employed Decision Tree and Random Forest algorithms for phishing website detection. Their findings demonstrated that machine learning methods can achieve high classification accuracy; however, they still face operational challenges when handling dynamic URLs and URL shortening services, which often trigger high false-positive rates. From a software engineering perspective, several studies have adopted the Software Development Life Cycle (SDLC) approach. The Waterfall model has been widely used in the

development of phishing detection systems due to its systematic and structured stages, making it highly suitable for web-based application development.

Based on the reviewed studies, it can be concluded that most existing research focuses on machine learning, deep learning, or rule-based methods independently. Rule-based approaches are effective in identifying predictable phishing patterns, while AI-based approaches are better at recognizing complex and previously unseen patterns. However, there is still limited research that combines the structural strengths of rule-based detection with the advanced contextual reasoning capabilities of Large Language Models (LLMs). Therefore, this study proposes a hybrid approach that integrates rule-based detection and LLM analysis to develop a phishing detection system that is more adaptive, systematic, interpretable, and accurate in identifying both known and emerging phishing patterns.

Table 2.1 Literature Review

No.	Research Title	Author / Year	Methods Used	Cons	Synthesis
1	Web-Based Phishing Link Detection System	Harahap et al. (2024)	Random Forest	Not yet able to detect new phishing patterns adaptively	Machine Learning
2	Phishing Detection Algorithms	Windarni et al. (2023)	Naïve Bayes, Decision Tree, Random Forest	Focus only on machine learning classification	Machine Learning
3	Web-Based Malicious URL Classification	Adiputra and Setiawan (2023)	Improved Random Forest	Haven't used contextual analysis yet	Machine Learning
4	Phishing URL Detection Using XGBoost	Lukito and Handaya (2022)	XGBoost, Random Forest	Haven't integrated rule-based methods yet	Machine Learning
5	Phishing URL Detection Using NLP	Aryanti and Nabila (2023)	SVM + TF-IDF	Relying on text representation	Machine Learning & NLP
6	Character-Level CNN for Phishing Detection	Aliofey et al. (2020)	Deep Learning (CNN)	Requires high computing	Deep Learning
7	Rule Based Attribute Checking	Pradana et al. (2019)	Rule-Based	Unable to detect unpredictable phishing	Machine Learning

8	Lexical URL Analysis for Phishing Detection	Ma et al. (2011)	Statistical URL Analysis	Less flexible to new patterns	Machine Learning
9	A Framework for Detection and Measurement of Phishing Attacks	Garera et al. (2007)	Infrastructure Features & Page Relations	Highly dependent on real-time third-party network lookups	Machine Learning
10	Lexical URL Analysis for Discriminating Phishing Web Sites	Khonji and Iraqi (2011)	Lexical URL Analysis & Component Tokenization	Drop in accuracy if phishing sites are hosted on compromised legal domains	Statistical Analysis
11	The Impact of Public Information on Phishing Attack and Defense	Moore and Clayton (2008)	Empirical Analysis of Attack Ecosystems & Server Logs	Focuses on macro-level behavior, lacks automated real-time detection algorithm	Empirical Analysis
12	Phishing Detection System Using Random Forest and Advanced Machine Learning Approaches	Noori et al. (2023)	Supervised Machine Learning (Random Forest)	Vulnerable to advanced text obfuscation and unseen semantic shifts	Machine Learning
13	Detection of Phishing Websites Using Machine Learning Algorithms	Salvi et al. (2021)	Decision Trees & Random Forest	High false-positive rates with dynamic URLs or URL shorteners	Machine Learning
14	DESIGN AND IMPLEMENTATION OF WEB-BASED PHISHING DETECTION SYSTEM USING RULE-BASED AND LLM APPROACHES	Muhammad Aswin Jalu 2026	Rule-Based & Large Laguage Model (LLM)		Hybrid (Rule-Based + LLM)

2.2 Problem Statement

Phishing attacks continue to be a significant cyber threat with attack patterns that are increasingly complex and difficult to recognize by ordinary users. Although various detection systems have been developed, there are still limitations in the ability to detect new phishing patterns that do not follow previously known patterns. Previous research has shown that rule-based methods (Rule-based) is effective at detecting predefined phishing indicators, but has limitations in recognizing new patterns that are not included in the system rules. On the other hand, the method based on Machine Learning and AI have a better ability to recognize complex patterns, but generally require large data training processes and more complex implementation.

In addition, most previous studies still used separate detection methods, either using rule-based or machine learning. In fact, the combination of rule-based methods and Large Language Model (LLM) has the potential to produce a more effective phishing detection system because it is able to handle predictable phishing patterns and unpredictable phishing patterns. Therefore, this study focuses on the development of a phishing detection system based on a hybrid approach that combines rule-based detection and LLM to produce a more adaptive, systematic, and accurate phishing identification process.

2.3 Theoritical Foundations

2.3.1 Phishing

Phishing is a form of online fraud that aims to obtain sensitive user information, such as usernames, passwords, credit card details, banking data, and other personal information by impersonating a legitimate and trusted entity in electronic communications. Phishing attacks generally use internet media, such as fake emails, instant messaging, social media, and fake websites designed to resemble real sites to manipulate victims into voluntarily giving up their personal information. Phishing attacks usually use spoofing techniques by impersonating the identity of an official organization, such as a bank, email service, marketplace, or social media platform.

In addition, phishing also often takes advantage of an element of urgency or threat, for example by providing a warning that a user's account will be blocked if they do not verify their data immediately. This technique aims to encourage victims to act quickly without

conducting further checks on the authenticity of the information received. One of the main characteristics of phishing is the use of suspicious URLs that redirect users to fake sites. These URLs often have a similar structure to the original domain, but are modified using certain techniques, such as the use of IP addresses, the addition of special symbols, or the excessive use of subdomains to hide the primary domain. Therefore, analysis of the URL structure is one of the important methods in detecting phishing attacks.

2.3.2 Characteristics and Structure of URLs

A Uniform Resource Locator (URL) is a unique address used to identify resources on the internet. The basic structure of a URL consists of several main components, namely a schema or protocol (http/https), a subdomain, a second-level domain, a top-level domain (TLD), ports, paths, queries, and fragments. In the context of cybersecurity, URL structure can be used as an early indicator to detect phishing sites without having to access the content of a web page directly. Phishing attackers generally manipulate URL structures to trick users into thinking the site is legitimate. One of the techniques that is often used is the use of IP addresses instead of domain names, such as http://192.168.1.1/login. In addition, the "@" symbol in the URL is also often used to hide the true purpose of the link.

URLs that are too long can also be an indication of phishing because they are often used to disguise suspicious parts of the domain. Other characteristics that are common in phishing URLs are the excessive use of hyphens in domain names, too many subdomains, and the presence of sensitive keywords such as login, secure, verify, account, and banking. The use of a URL shortener service can also be used to hide the actual destination address, increasing the risk of phishing. Analysis of URL characteristics and structures is important because this method is able to detect indications of phishing quickly and efficiently without the need for a more complex process of analyzing the content of a web page.

2.3.3 Rule-Based Detection

Rule-based detection (Rule-based) is an approach in cybersecurity that works based on a predefined set of rules and indicators. In the context of phishing detection, this method is used to analyze the characteristics of URLs and compare them to known phishing patterns. Method Rule-based works using IF–THEN logic. For example, if a URL uses a numeric IP address, then it can be categorized as suspicious. Other commonly used rules include the presence of the "@" symbol, the length of the URL that exceeds a certain limit, the use of many subdomains, the use of hyphens on domains, and the presence of sensitive keywords

such as Login, Secure, and Verify. This approach has several advantages, namely a fast and efficient detection process because it only matches the rules that are already available.

In addition, the rule-based method is also transparent because the rules used are easy to understand and explain, thus facilitating the process of auditing and evaluating the system. This method is very effective for detecting common and predictable phishing patterns. However, rule-based methods also have limitations. It is difficult for systems to detect new phishing patterns that have never been defined in the rules (zero-day phishing). In addition, the rules must be constantly updated to keep up with the development of phishing attack techniques. If the rules are too strict, the system can produce a phishing site that is considered safe when it is actually dangerous (false positive), while rules that are too loose can cause a phishing site to be considered safe when it is actually dangerous (false negative).

2.3.4 Large Language Model (LLM)

Large Language Model (LLM) is an artificial intelligence model that is trained using large amounts of text data to understand, process, and generate human language. LLMs have the ability to recognize complex patterns, understand the context of language, and conduct in-depth analysis of information structures. In this study, LLM was used as a follow-up analysis to detect phishing that could not be recognized by rule-based methods. The LLM serves to understand the relationships between indicators so that the system not only relies on rule scores, but is also able to analyze URL patterns thoroughly. The application of LLM in phishing detection includes analyzing URL structures, identifying domain manipulation patterns, analyzing the context of keyword usage, and evaluating the similarity of URLs to official domains. Unlike rule-based methods that only check certain indicators, LLMs are able to understand the relationships between elements in URLs so that they better adjust to the conditions in detecting new phishing patterns (unpredictable phishing).

In this study, the model used is GPT-4o-mini The model receives input in the form of URLs and initial analysis results from the rule-based engine. Furthermore, the LLM conducts a thorough analysis to produce a final classification in the form of a safe or phishing category along with an explanation of the analysis results. LLMs have several advantages, such as the ability to understand analysis in depth, adapt conditions to new attack patterns, and be able to provide a more comprehensive understanding of the characteristics of phishing URLs. However, the use of LLMs also has drawbacks, including

requiring larger computing resources and a more complex analysis process than ordinary heuristic methods.

2.3.5 Hybrid Decision Making Model

The hybrid decision-making model is an approach that combines rule-based methods (Rule-based) with Large Language Model (LLM) in the process of phishing detection. This approach is used to take advantage of the advantages of each method to produce a more accurate, adaptive, and interpretive detection system. In this model, the Rule-based used as an initial stage to detect known or predictable phishing indicators (predictable phishing), such as the use of IP addresses, the "@" symbol, unnatural URL lengths, the use of sensitive keywords, the excessive number of subdomains, and the use of hyphens on domains. This process is done by matching the URL structure to a predefined rule so that it can generate a quick and systematic initial classification.

The results of the initial rule-based analysis are then forwarded to the LLM module for a thorough follow-up evaluation. Although the system uses a hybrid approach, the decision-making process still adjusts to the conditions of the URL being analyzed. If the phishing indicators found are in accordance with the available rules, then the results of the rule-based analysis can be used as the basis for an initial decision. However, if the URL does not have a clear indicator or contains a pattern that has not been defined in the rules (unpredictable phishing), the LLM will conduct a more in-depth analysis by evaluating the overall pattern and context of the URL.

LLM functions as a final decision maker who is able to understand the relationship between indicators thoroughly so that the system does not rely solely on rule scores. With this approach, the system is able to detect new phishing patterns that are not recognizable by rule-based methods, while helping to reduce the likelihood of false detection errors such as false positives and false negatives. The integration between rule-based methods and LLMs results in a phishing detection system that is more flexible, adaptive, and able to adapt to the evolving development of phishing attack techniques.

CHAPTER III
METHODOLOGY

3.1 Research Methods

This study uses a Research and Development (R&D) method, considering that its main focus is not only limited to theoretical analysis, but also on the development of a web-based phishing detection system software product. To ensure the development process runs in a structured and systematic manner, this study adopts the Waterfall Software Development Life Cycle (SDLC) model consisting of five sequential stages: requirements analysis, system design, implementation, testing, and maintenance and evaluation. Through the requirements analysis phase, phishing indicators and integration between the Rule-Based and Large Language Model (LLM) modules are determined, which is then followed by the system architecture design. The implementation phase includes the development of frontend and backend interfaces that handle URL analysis and communication with GPT-4o-mini, while implementing rule-based detection for indicators such as IP address usage, URL length, and URL shortener usage. After that, a testing phase is conducted to validate the entire system functionality, which concludes with a maintenance phase to optimize performance and contextual analysis capabilities. By implementing the Waterfall model within the R&D framework, the development process becomes more efficient and directed, resulting in a phishing detection system capable of providing more adaptive, interpretative, and comprehensive identification results.

3.2 Requirements

3.2.1 Hardware Requirements Analysis

The hardware used in this study includes:

- Laptops with specifications
  - Processor : AMD RYZEN 5 6000 SERIES
  - RAM : 16 GB
  - Storage : 500 GB
- Internet connection for LLM access and system testing

3.2.2 Software Needs Analysis

The software used includes:

- Operating System: Windows 10/11, Linux, or macOS
- Development Environment & Runtime: Node.js (v24.13.0)
- Backend Framework: Express.js
- Database Management System: MySQL 9.7.1
- Containerization Platform: Docker Platform
- Web Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge

3.2.3 System Needs Analysis

The designed system has the following flow:

- Input

User-entered URLs

- Process

1. Valdiasi URL
2. Rule-Based Analysis
3. Risk score calculation
4. LLM Analysis
5. Final decision making

- Output

1. URL Classification (Secure/Phishing)
2. Risk level
3. Analysis explained
4. PDF Report

3.3 System Designs

The system is designed to describe the system's workflow in analyzing URLs provided by the user. In general, the system developed uses a hybrid detection approach, which is a combination of rule-based analysis and Large Language Model (LLM)-based analysis. The analysis process begins when the user enters the URL into the system. URLs are then analyzed using rule-based methods to detect phishing indicators based on the characteristics of the URL structure. Some of the rules used in the analysis include detecting the use of IP addresses instead of domains, detecting the use of the "@" symbol in URLs, URL length analysis to detect URLs that are too long, analyzing the presence of sensitive keywords that are often used in phishing attacks, short URL analysis to detect URLs that are too short, detecting too many

subdomains, detecting hyphens (-) on a domain. Each indicator found will add to the risk score. The results of this analysis then result in an initial classification in the form of a URL risk level.

Furthermore, the results of the rule-based analysis are sent to the LLM module to see not only the number of scores and indicators, but to clarify the meaning and relationship of the Rule-based and Large Language Model (LLM). The LLM will analyze the indicators found and provide an explanation of potential phishing on the URL. The result of this process is the final decision on the status of the URL, whether it is classified as safe or potentially phishing. In addition, the system also stores the analysis results in a database and provides a feature to export the analysis report in the form of a PDF document.

3.3.1 System Architecture

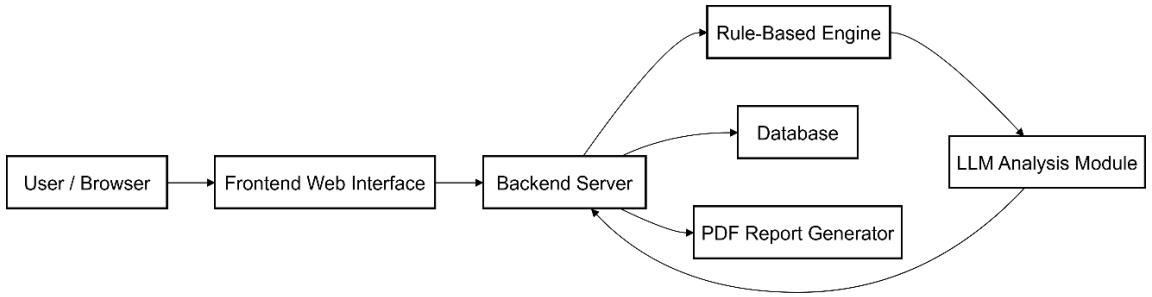


Figure 3.1System Architecture.

Figure 3.1 In System Architecture, it maps out the main components that make up the system. The user interacts through an interface (Frontend), which sends the URL to the Backend Server. The server then distributes the tasks to the Rule-Based Engine and the LLM Module. The results of the analysis will be stored in the Database and can be printed as a report through the PDF Report Generator.

3.3.1.1 User/Browser

Users access the system through a web browser, users enter the URL they want to analyze on the web page, view the results of the analysis, and download the report in PDF format.

3.3.1.2 Frontend Web Interface

It is a web interface built using HTML, CSS, and JavaScript. This section accepts URL input from the user, displays the analysis process in real-time, presents the results of the analysis (rule-based, LLM, and Final decision), and provides a button to download PDF reports.

3.3.1.3 Backend Server : Rule-Based Engine, Database, PDF Report Generator

The backend server serves as a system control center that receives URL input from the frontend and distributes the analysis process to several key components of the system. The backend is developed using Node.js and run using Docker to maintain the consistency of the system environment and simplify the application implementation and deployment process. The main components of the Server Backend consist of :

- **Rule-Based Engine**

  The Rule-Based Engine functions to process the URLs entered by the user and conduct risk assessments based on predefined rules. The analysis was conducted using several phishing indicators, such as the use of IP addresses, the "@" symbol, URL length, sensitive keywords, the number of subdomains, the use of URL shorteners, and hyphens on the domain. This process runs on a Docker container so that the system can run in a stable, isolated, and easily configurable manner.

- **Database**

  The database functions to store data from system analysis, including URLs, rule-based scores, LLM analysis results, risk categories, and analysis time (timestamp). Data storage is done to support the reporting process and analysis history.

- **PDF Report Generator**

  PDF Report Generator is used to create analysis reports in PDF format based on data stored in the database. The report contains the results of phishing detection, risk categories, indicators found, and analysis results from LLMs.

3.3.1.4 LLM Analysis Module

The Large Language Model (LLM) Analysis Module is an advanced component used in the proposed hybrid phishing detection system. This module works together with the Rule-Based Detection module to improve the accuracy and adaptability of phishing identification. While the Rule-Based module is responsible for detecting predefined phishing indicators, the LLM module performs contextual analysis to provide a more comprehensive evaluation of the analyzed URL.

In this reasearch, the GPT-4o-mini model is utilized to analyze the results generated by the Rule-Based Detection module. The input received by the LLM includes the analyzed URL, detected phishing indicators, risk score, and the initial

classification produced by the rule-based engine. Based on this information, the LLM evaluates the relationship among indicators, examines the overall URL structure, and interprets phishing patterns that may not be fully represented by predefined rules.

Unlike a standalone LLM approach, the proposed hybrid method does not use the LLM as a replacement for the Rule-Based module. Instead, the LLM functions as an advanced analysis layer that validates, interprets, and enhances the initial assessment generated by the Rule-Based Detection module. This integration enables the system to combine the speed and consistency of rule-based analysis with the contextual reasoning capability of the LLM.

The hybrid workflow begins with the Rule-Based Detection module, which extracts phishing-related indicators such as the use of IP addresses, the presence of the "@" symbol, abnormal URL length, sensitive keywords, URL shorteners, excessive subdomains, and hyphen usage within the domain. The resulting risk score and detected indicators are then forwarded to the LLM module for further evaluation. The LLM analyzes the relationships among these indicators and assesses the URL pattern as a whole before producing the final classification.

The final output generated by the LLM consists of a security classification, namely Safe, Low Risk, Not Recommended, or Phishing, accompanied by a structured explanation of the decision. Through this hybrid approach, the system is able to detect both predictable phishing patterns identified by predefined rules and more complex phishing patterns that require contextual interpretation, thereby improving the effectiveness of phishing detection compared to using a single detection method.

A. Prompt Design and Model Configuration

The LLM component was implemented using the GPT-4o-mini model through the OpenAI API. The model was configured with a temperature value of 0.2 to minimize response variability and a JSON response format to ensure that the generated output could be processed automatically by the backend system.

The system prompt defines the LLM as a cybersecurity and digital forensics assistant responsible for contextual phishing URL analysis. The prompt restricts the classification output to four system-defined risk levels: Safe, Low Risk, Not Recommended, and Phishing. The LLM is also instructed not to generate the original dataset labels, such as Benign, Defacement, or Malware, because those labels are normalized separately during the evaluation process.

In hybrid analysis, the user prompt contains the analyzed URL, Rule-Based score, and detected indicators. The LLM uses this information to provide contextual interpretation and generate a final decision, predicted label, confidence score, and explanation. The backend validates the generated decision and confidence score before returning the result to the frontend. If the LLM is unavailable or returns an invalid response, the system uses the Rule-Based classification as a fallback mechanism.

Tabel 3.1 GPT-4o-mini Promt Implementation

```
app.post("/analyze", async (req, res) => {
  try {
    const {
      url,
      ruleScore = 0,
      rules = [],
      mode = "hybrid"
    } = req.body;

    if (!url) {
      return res.status(400).json({
        error: "URL wajib disertakan."
      });
    }

    const systemInstruction = `
You are an AI assistant specialized in digital forensics,
cybersecurity, and phishing URL analysis.

Return ONLY a valid JSON object.

The system has exactly four risk classifications:

1. Safe
2. Low Risk
3. Not Recommended
4. Phishing

Definitions:

Safe:
The URL does not show significant suspicious indicators.

Low Risk:
The URL contains minor suspicious characteristics,
but the indicators are not sufficient to classify it
as a significant threat.

Not Recommended:
The URL contains significant suspicious or malicious
characteristics and should not be trusted.

Phishing:
The URL strongly indicates phishing, credential theft,
impersonation, fraudulent login pages, or social engineering.
```

```
The "decision" field MUST contain exactly one of:
"Safe", "Low Risk", "Not Recommended", "Phishing"

The "predicted_label" field MUST contain exactly one of:
"Safe", "Low Risk", "Not Recommended", "Phishing"

DO NOT output:
"benign"
"defacement"
"malware"

Return exactly:

{
  "decision": "Safe",
  "predicted_label": "Safe",
  "confidence_score": 0,
  "reason": "Brief explanation in English."
}
`;

    let userContent = `URL: ${url}\n`;

    if (
      mode === "hybrid" ||
      mode === "hybrid-full" ||
      mode === "hybrid-bypass"
    ) {
      userContent +=
        `Rule-Based Score: ${ruleScore}\n`;

      userContent +=
        `Rule-Based Indicators: ${
          Array.isArray(rules) &&
          rules.length > 0
            ? rules.join(", ")
            : "None"
        }\n`;
    }

    if (mode === "llm") {
      userContent += `
Perform an independent contextual
analysis of the URL.
`;
    }

    const completion =
      await openai.chat.completions.create({
        model: "gpt-4o-mini",

        messages: [
          {
            role: "system",
            content: systemInstruction
          },
          {
            role: "user",
```

```
            content: userContent
          }
        ],

        temperature: 0.2,

        response_format: {
          type: "json_object"
        }
      });

const rawContent =
  completion?.choices?.[0]?.message?.content;

if (!rawContent) {
  throw new Error(
    "OpenAI tidak mengembalikan content."
  );
}

const parsed =
  JSON.parse(rawContent.trim());

let finalDecision =
  parsed.decision;

const validDecisions = [
  "Safe",
  "Low Risk",
  "Not Recommended",
  "Phishing"
];

if (
  !validDecisions.includes(
    finalDecision
  )
) {
  finalDecision =
    getRiskCategory(ruleScore);
}

let confidence =
  Number(
    parsed.confidence_score
  );

if (Number.isNaN(confidence)) {
  confidence = 0;
}

confidence =
  Math.max(
    0,
    Math.min(
      100,
      confidence
    )
  );
```

```
      res.json({
        finalDecision,
        llmAnalysis:
          parsed.reason ||
          "No explanation provided.",
        predictedLabel:
          parsed.predicted_label ||
          finalDecision,
        confidenceScore:
          confidence
      });

    } catch (err) {

      const fallbackDecision =
        getRiskCategory(
          req.body?.ruleScore || 0
        );

      res.status(200).json({
        finalDecision:
          fallbackDecision,

        llmAnalysis:
          "LLM gagal diproses. " +
          "Rule-Based digunakan sebagai fallback.",

        predictedLabel:
          fallbackDecision,

        confidenceScore: 0,

        fallback: true
      });
    }
});
```

The prompt consists of two primary inputs, namely the URL being analyzed and the Rule-Based indicators generated during the initial detection process. In hybrid analysis, the Rule-Based score is also provided to the LLM as additional contextual information. These inputs are combined into the user prompt and submitted to GPT-4o-mini for contextual evaluation.

The model then generates four types of information: decision, predicted label, confidence score, and reason. The decision and predicted label are restricted to the four risk categories defined by the application: Safe, Low Risk, Not Recommended, and Phishing. This restriction ensures consistency between the LLM output and the classification categories used by the application.

The backend also performs validation after receiving the response. If the generated decision is invalid or the LLM cannot process the request, the system uses the Rule-Based classification as a fallback mechanism.

Tabel 3. 2 Risk Classification Function

```
function getRiskCategory(score) {
  const numScore = Number(score) || 0;

  if (numScore === 0) {
    return "Safe";
  }

  if (numScore >= 1 && numScore <= 3) {
    return "Low Risk";
  }

  if (numScore >= 4 && numScore <= 6) {
    return "Not Recommended";
  }

  return "Phishing";
}
```

In the design of this system, the results of the URL analysis do not simply display technical dataset labels; instead, they are translated into four risk levels that are easily understood by users: Safe, Low Risk, Not Recommended, and Phishing. These risk levels represent the threat severity based on weightings from a rule-based engine that encompasses the full range of standard cybersecurity dataset classes (Benign, Phishing, Defacement, and Malware).

Tabel 3.3 GPT-4o-mini Configuration Parameters

Parameter	**Configuration**
Model	GPT-4o-mini
Provider	OpenAI API
Temperature	0.2
Response Format	JSON Object
Input	URL, Rule-Based Score, Rule-Based Indicators
Output	Decision, Predicted Label, Confidence Score, Reason
Risk Classes	Safe, Low Risk, Not Recommended, Phishing
Fallback	Rule-Based Classification

Table 3.3 presents the configuration parameters of the GPT-4o-mini model used in the proposed phishing detection system. The model operates as the contextual analysis component within the hybrid architecture. The temperature value of 0.2 is selected to reduce response variability and encourage consistent classification results. The output is configured using the JSON object format to facilitate structured processing by the backend.

The model receives the URL as the primary input. In hybrid mode, the Rule-Based score and detected indicators are also provided as supporting information. The generated response consists of a classification decision, predicted label, confidence score, and explanatory reasoning.

The four risk categories used by the LLM are Safe, Low Risk, Not Recommended, and Phishing. These categories are intentionally consistent with the Rule-Based classification function so that the outputs of both components can be integrated within the same decision pipeline.

A. Promt Engineering Architecture

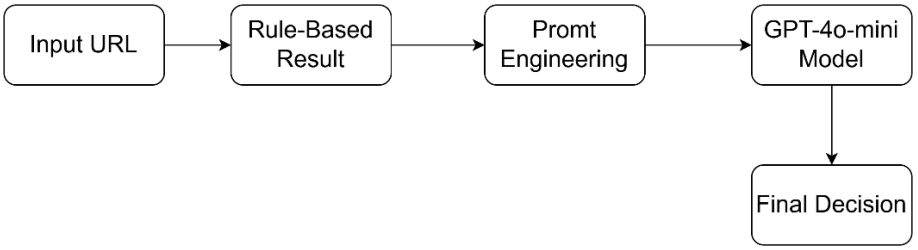


Figure 3. 2 Prompt Engineering Workflow in Hybrid Method

Figure 3.2 illustrates the specific architecture of the Prompt Engineering process within the proposed system. The workflow begins when the user provides an Input URL. The URL is first statically analyzed to produce a Rule-Based Result, which contains a threat score and technical indicators. Instead of presenting this intermediate result directly, the system passes it into the Prompt Engineering phase. During this phase, the rule-based findings are dynamically woven into a structured System Prompt, providing the LLM with specific threat contexts. The enriched prompt is then dispatched to the GPT-4o-mini model, which performs the final reasoning to produce the Classification and narrative explanation regarding the safety of the URL.

3.4 Workflow of the Work System

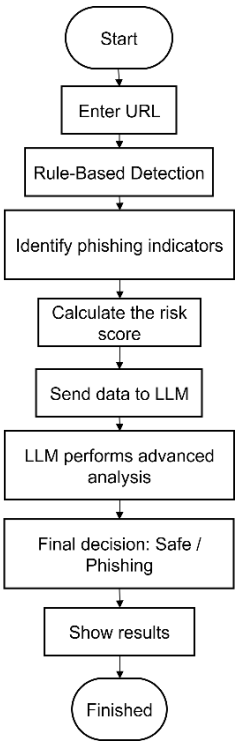


Figure 3.3 Workflow of the Work System.

Figure 3.3 The system workflow starts when the user enters the URL through the web interface. The URL is sent to the backend to be processed using rule-based methods to detect early indicators of phishing. The system then calculates the risk score based on predetermined rules. Furthermore, the results of the rule-based analysis are sent to the LLM module for a more in-depth thorough analysis. The LLM will evaluate the combination of indicators found and provide a final decision regarding the status of the URL. The final results in the form of URL security classification, risk level, and analysis explanation will be displayed to the user and stored in the database. The system also provides a report creation feature in PDF format

3.5 Rule-Based System

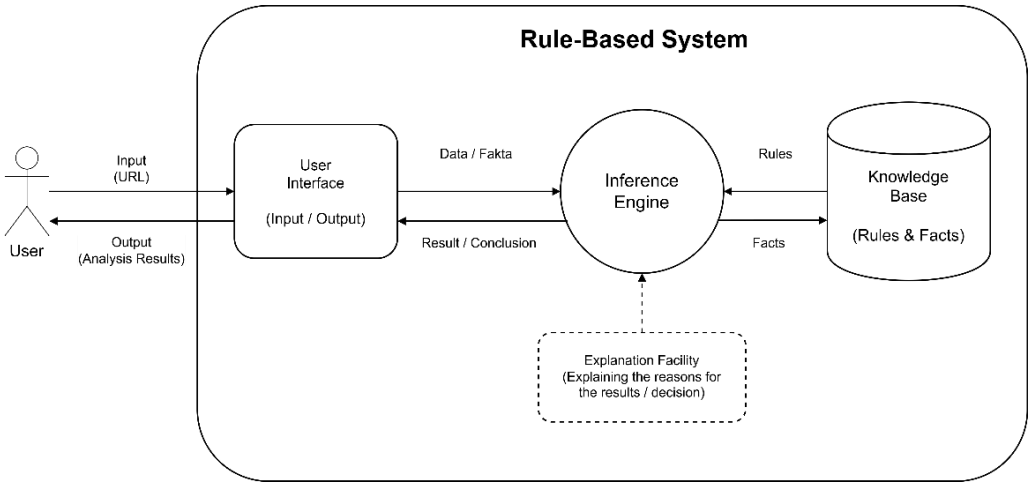


Figure 3. 4 Rule-Based System

Figure 3.4 shows the Rule-Based System architecture used to carry out the analysis and decision-making process based on certain predetermined rules. This system works by utilizing the relationship between input data, knowledge base, and inference engines to produce a decision or conclusion. In phishing detection research, a rule-based system is used to analyze the characteristics of URLs and determine whether a URL is safe, suspicious, or phishing based on the rules that have been created. The process starts from the User Interface section, which is the interface that the user uses to enter input in the form of the URL of the website to be analyzed.

In addition to receiving input, this section also functions to display the results of the analysis generated by the system to the user. Once the URL is entered, the data is sent to the Inference Engine section as the main processing center in a rule-based system. The Inference Engine section functions to carry out the reasoning and analysis process of the data received from users. The inference engine will compare the data or facts obtained from the URL with the rules stored in the Knowledge Base. In the context of phishing detection, the rule can be suspicious URL characteristics, such as the use of numeric IP addresses, excessive URL lengths, abnormal number of subdomains, or the use of manipulative words such as "login", "verify", and "secure".

The inference engine then performs an evaluation based on the appropriate rules to determine the risk level of the analyzed URL. The Knowledge Base section is a knowledge base that contains a collection of rules and facts used in the decision-making process. The knowledge base is the main source of information for the system to determine whether or not a URL has an indication of phishing. The more complete the rules have, the system's ability to recognize phishing patterns can also increase. However, these rules need to be updated regularly to be able to keep up with the latest phishing techniques. After the analysis process is completed, the results of the inference engine are sent back to the User Interface in the form of outputs or analysis results.

The output can be a URL classification such as safe, suspicious, or phishing, along with indicators detected during the analysis process. In addition, the diagram also has an Explanation Facility section, which is a component that functions to provide an explanation of the reason why the system produces a decision. This facility helps users understand what indicators or rules cause a URL to be categorized as phishing, making the analysis results more transparent and easy to understand.

3.5.1 Categories Rule-Based

Tabel 3.4 List Rule-Based

Rule	IF THEN
Rule 1	If the url is in the form of an ip adress then phishing is high
Rule 2	If there is an @ symbol then phishing is high
Rule 3	If the URL is >75 characters long, then phishing risk is medium
Rule 4	If the url is <75 characters long, then phishing is safe
Rule 5	If the password is sensitive to login, then phishing risk is low
Rule 6	If the keyword is sensitive verified, then phishing risk is low
Rule 7	If the password is sensitive secure, then phishing risk is low
Rule 8	If the keyword is sensitive to updates then phishing risk is low
Rule 9	If the password is account sensitive then phishing risk is low
Rule 10	If the keyword is banking sensitive then phishing risk is low
Rule 11	If the password is wallet sensitive then phishing risk is moderate
Rule 12	If the password is password sensitive then phishing risk is low
Rule 13	If the keyword is sensitive support then phishing risk is low
Rule 14	If the keyword is sensitive free , then phishing risk is low

Rule 15	If the keyword is admin sensitive then phishing risk is low
Rule 16	If the keyword is sensitive to the service , then phishing risk is low
Rule 17	If the use of URL Shorteners bit.ly then phishing is moderate
Rule 18	If the use of URL Shortener tinyurl.com then phishing is moderate
Rule 19	If the use of URL Shortener s.id then phishing is moderate
Rule 20	If the number of detected subdomains is more than 3 dots (.) in the hostname (without HTTP/HTTPS) then phishing is moderate
Rule 21	If there is a hyphen (-) in the name of the hostname or domain then the phishing risk is low

Table 3.4 shows the rule-based rules used by the system to initially identify the analyzed URLs. Each rule is designed based on characteristics commonly found on phishing sites. The system examines the URL against these rules and assigns a risk score based on the detected indicators. The first to third rules focus on URL structure characteristics, such as the direct use of IP addresses, the presence of the "@" symbol, and URLs exceeding 75 characters in length. The use of IP addresses is often found on phishing sites because perpetrators attempt to conceal the domain's true identity. The "@" symbol is also often used to trick users by obscuring the URL's true purpose. Furthermore, excessively long URLs are commonly used to conceal suspicious parts of the domain.

The fifth to sixteenth rules relate to the presence of sensitive keywords such as login, verify, secure, update, account, banking, wallet, password, support, free, admin, and service. These words are often used by phishers to build user trust or encourage users to take specific actions, such as entering personal information or account credentials.Rules seventeen through nineteen detect the use of URL shortening services such as bit.ly, tinyurl.com, and s.id. These services are often exploited to conceal the actual destination address, increasing the risk of phishing. Furthermore, rule twenty detects an excessive number of subdomains, i.e., more than three dots in the hostname, which can indicate domain impersonation. The final rule detects the use of hyphens in domain names, as this pattern is often found in domains impersonating legitimate sites.

Tabel 3. 5 List of risk level classifications based on Rule-Based scores

Total Score (Rule Score)	Risk Level Category	Initial System Assessment	Description of Indications
0	Safe	Secure	No phishing indicators or structural anomalies were detected on the URL
1 – 3	Low Risk	Need Attention	Found few suspicious indicators at mild levels that require caution
4 – 6	Not Recommended	Suspicious	Found strong fraud indicators or a combination of many mild indicators
>6	Phishing	Critical Risk	The threat level is very strong and the chances are very high that the URL is a phishing site

Table 3.5 shows the risk level classification mechanism based on the total score obtained from the rule-based analysis process. Each detected indicator is assigned a specific score, and all scores are then summed to produce a final score, called the Rule Score. If the score is 0, the URL is categorized as Safe because no phishing indicators or anomalies were found in the URL structure. If the score is between 1 and 3, the URL is categorized as Low Risk, indicating the presence of several mild suspicious indicators, requiring user vigilance. Scores of 4 to 6 are categorized as Not Recommended because the system detects a combination of several fairly strong phishing indicators. Meanwhile, scores above 6 are categorized as Phishing, indicating a very high probability that the URL is malicious and could potentially harm users. This classification serves as an initial analysis result before further evaluation by the Large Language Model (LLM) module.

Tabel 3. 6 Characteristics and weights of phishing detection

NO	Characteristics	Conditions / Criteria	Score Weights
1	Use of IP Addresses	URLs use the IP format (192.168.x.x) instead of the domain	3
2	The Presence of the "@" Symbol	There is an "@" character inside the URL	3
3	URL length	The number of URL characters exceeds the 75-character limit	1
4	Sensitive Keywords	There are passwords such as (login, verify, secure, update, account, banking, wallet, password, support, free, admin, service)	1 (per keyword)
5	Use of URL Shortener	Using link shortening services (bit.ly, tinyurl.com, s.id)	2
6	Number of Subdomains	More than 3 dots (.) detected inside the hostname (without HTTP/HTTPS)	2
7	Use of hyphens (-)	There is a hyphen (-) on the hostname or domain naming	1

Table 3.6 explains the characteristics used in the phishing detection process and the weighting assigned to each indicator. Weighting is used to indicate the level of influence a characteristic has on the likelihood of phishing. The characteristics with the highest weighting are the use of IP addresses and the presence of the @ symbol, each with a weighting of 3 points because these two indicators are frequently found in phishing URLs and carry a relatively high risk level. The use of URL shortening services and an excessive number of subdomains are given a weighting of 2 points because they are often used to disguise the site's true identity.

Meanwhile, indicators such as excessive URL length, the use of sensitive keywords, and the use of hyphens in domains are given a weighting of 1 point because, although frequently found in phishing sites, these indicators can also appear on legitimate sites, thus carrying a relatively lower risk level. Through this weighting, the system can calculate the risk level more proportionally based on the number and severity of indicators found in the analyzed URL. The

resulting score is then used as the basis for the risk classification process and further analysis by the LLM module.

3.5.2 LLM Analysis

Large Language Model (LLM) was used in this study to detect phishing that cannot be recognized by rule-based methods. LLMs act as an advanced analysis component that is able to understand URL patterns in a more in-depth and thorough manner. The module uses the GPT-4o-mini model to analyze indicators generated by rule-based engines, both predictable and unpredictable indicators. The analysis is carried out adaptively according to the conditions of the analyzed URL. The LLM receives input in the form of URLs as well as the initial analysis results from the rule-based.

Furthermore, LLM conducts a follow-up analysis by understanding the meaning and relationships between indicators as a whole. With this approach, the LLM not only relies on the number of indicator scores, but also considers the following URL structure patterns, similarities to the original domain, manipulation techniques commonly used in phishing attacks. In its implementation, if the rule-based engine finds a strong indicator according to a predetermined rule, then the results can be used as the basis for the initial decision.

However, if rule-based does not find significant indicators or generate safe conditions, the LLM will conduct a more in-depth analysis independently without relying on the indicator score, by evaluating the overall URL pattern. As such, LLMs are able to detect unpredictable phishing, which is an attack pattern that is not defined in the system's rules.

3.5.3 Final Decision

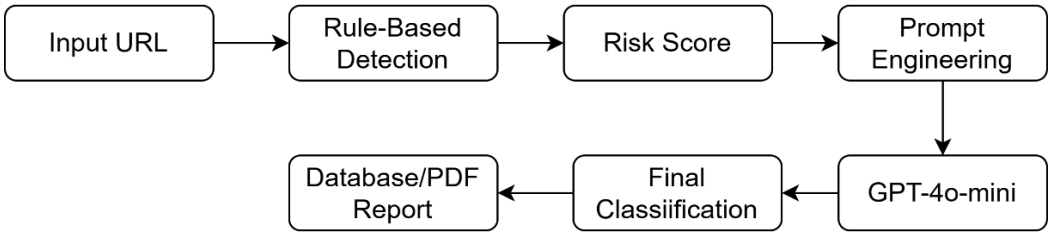


Figure 3.5 architecture of the hybrid model-based phishing detection system

The Final Decision stage is a crucial phase and the final chapter in the architecture of the hybrid model-based phishing detection system. Based on the system's workflow, the evaluation process begins when an input URL is entered into the system for processing by the Rule-Based Detection component to statically identify structural anomalies and phishing indicators. The test results from the rule-based detection engine then calculate a numeric Risk Score. This score

and the list of detected rule indicators are then combined into the Prompt Engineering process as a knowledge base (contextual background) before being sent to the GPT-4o-mini artificial intelligence module.

In this integration stage, the final system security decision is not simply derived from the rules engine's rigid mathematical calculations. Although the rules-based engine systematically groups risk levels into four standard clusters Safe for a total score of 0, Low Risk for a score range of 1–3, Not Recommended for a score range of 4–6, and Phishing for an accumulated score above 6 the system still leverages the power of Large Language Models (LLMs) to perform deeper contextual and interpretive analysis. Through LLM intervention, the system is able to evaluate correlations between indicator components, variations in semantic language patterns, social engineering, and advanced manipulation of URL strings that conventional rule-based inspections often fail to detect.

In this hybrid model, the LLM plays a central role as the final decision maker, capable of validating, interpreting, or even overriding previous rule-based classification results if stronger contextual anomalies are detected in the analyzed URL data. After a definitive decision regarding the URL's safety level is made by the LLM the output of which remains aligned to the categories Safe, Low Risk, Not Recommended, or Phishing the process proceeds to the Final Classification stage. As the final step of the entire detection cycle, all forensic data, including URL links, rule indicator track records, risk scores, and explanatory analysis narratives from the LLM, are executed, stored in a database, and exported as a formal report in the form of a Database/PDF Report.

3.5.4 Application Design

To design a web-based application in this study, a system design approach was used which included user interface design, use case, system architecture, and database design. The design was carried out to describe the system's workflow, the relationship between components, and the process of interaction between users and the developed phishing detection system. The user interface design is done to provide an easy-to-use system view in the process of entering URLs, viewing analysis results, and downloading detection reports. Furthermore, use cases are used to describe user activity and interaction with the system.

The design of the system architecture is used to show the relationship between the frontend, backend, rule-based engine, LLM module, database, and PDF report generator in the phishing analysis process. In addition, a database was also designed to store all data from phishing analysis, such as analyzed URLs, rule-based scores, LLM analysis results, final classification,

and timestamps. Databases are implemented using Docker to make it easier to manage services and maintain consistency in the system environment. With the design of the system, the application implementation process can be carried out in a more structured manner and in accordance with research needs.

1. Use Case

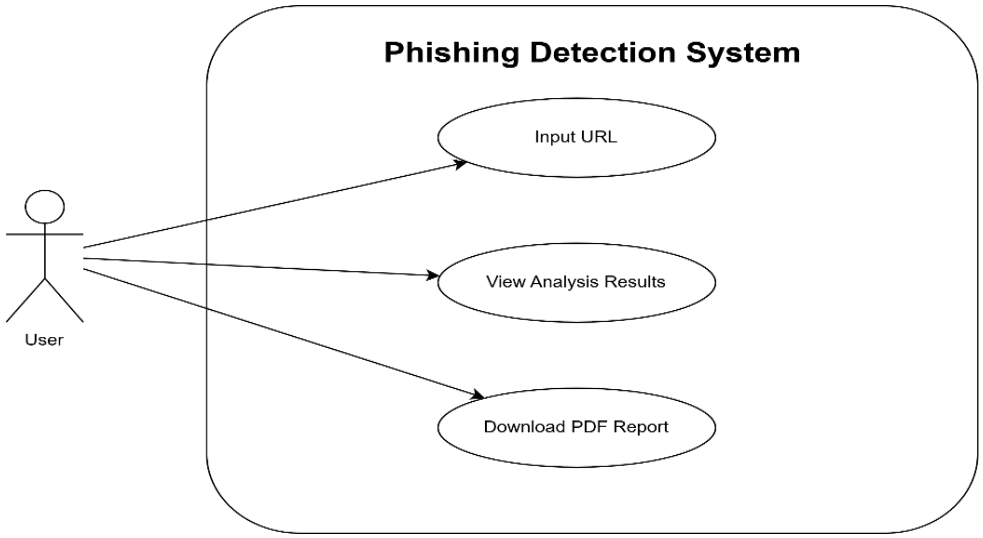


Figure 3.6 Phising Detection System

The Figure 3.6 Phising Detection System diagram illustrates the interaction between users with a web-based phishing detection system. Users can enter the URL to analyze, view the results of the phishing analysis, and download the analysis report in PDF format. The system then performs an analysis process using rule-based methods and LLM before saving the analysis results to the database.

2. Database Design

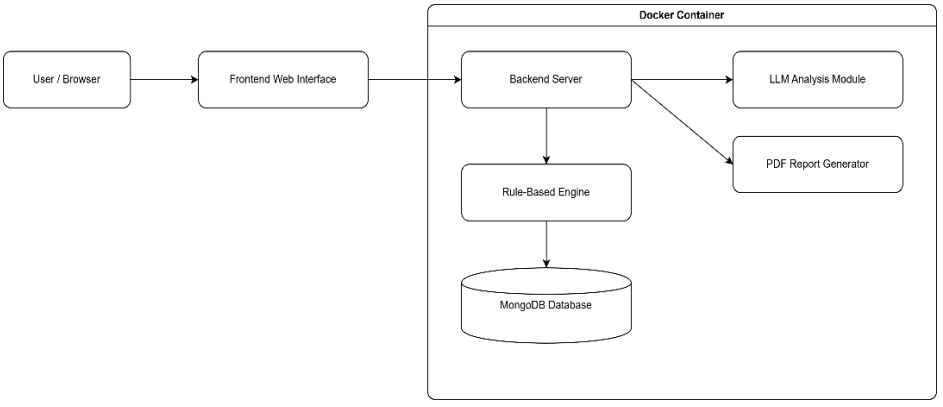


Figure 3. 7 MySQL 9.7.1

The Docker Container in Figure 3.7 diagram illustrates the implementation of a hybrid phishing detection system architecture within an isolated container environment. The workflow begins with the front-end user interface, where the user enters the target URL for forensic examination. The back-end server acts as a central controller, receiving network requests and orchestrating sequential analysis between the deterministic rule-based engine and the GPT-4o-mini contextual LLM module.

Once the final classification is determined, the analysis log, including URL strings, structural indicators, cumulative risk scores, and LLM natural language explanations, is permanently stored in a MySQL 9.7.1 database instance running within a dedicated container volume. Furthermore, the back-end integrates seamlessly with an automated PDF generation component to facilitate the compilation and export of formal forensic analysis reports. Implementing the system through Docker containers ensures cross-platform environmental consistency, isolates software dependencies, simplifies system deployment, and improves the operational stability of application services.

3.6 Implementation

The system is implemented web-based using both frontend and backend architectures. The frontend serves as a user interface that is used to enter URLs, display analysis results, and download phishing detection reports in PDF format. The frontend is developed using HTML, CSS, and JavaScript to support an interactive and easy-to-use system view. The backend serves as the data processing center and the main controller of the system. The backend handles the rule-based analysis process, integration with Large Language Model (LLM), data storage of analysis results, and the creation of PDF reports.

The backend is developed using Node.js and Express.js to support the communication process between the frontend, database, and LLM modules. At the analysis stage, the system receives input in the form of a URL from the user. The URL is then processed by a rule-based engine to detect phishing indicators based on predefined rules, such as the use of IP addresses, the "@" symbol, the length of the URL, sensitive keywords, the number of subdomains, the use of URL shorteners, and hyphens on the domain. The results of the initial analysis are then sent to the LLM module for further evaluation of the URL pattern as a

whole. The data from the analysis is then stored in a database to support the documentation and reporting process of the system.

In addition, the system also provides a feature for generating analysis report results in PDF format that contains URL information, rule-based detection results, LLM analysis results, final classification, and analysis time. In implementation, backend and database services are run using Docker to maintain the consistency of the system environment, simplify the deployment process, and improve the stability and management of application services.

3.7 Functionality Testing

Functionality testing was conducted to verify that all features implemented in the phishing detection system operated according to their intended functions. This testing employed the Black Box Testing method, where each feature was evaluated based on its input and expected output without examining the internal source code.

The testing process covered all major system functionalities, including URL input validation, rule-based phishing analysis, Large Language Model (LLM) analysis, hybrid classification, database storage, analysis history retrieval, PDF report generation, and communication between frontend and backend services. Each test case was executed by providing predefined inputs and observing whether the resulting output matched the expected behavior.

The functionality testing results indicate whether each component of the system operates correctly and supports the overall phishing detection process. A feature is considered successful if the actual output produced by the system is consistent with the expected output defined during system design.

Tabel 3.7 Feature Testing Secenario

No	Tested Features	Testing Scenarios	Input	Expected Output	Results
1	Input URL	The user entered a valid URL	https://google.com	URLs are accepted by the system	Successful
2	URL validation	The user enters a blank URL	-	System displays error message	Successful

3	Rule-Based Analysis	System analyzes URLs using rule-based	https://paypal-login-secure.com	The system displays phishing scores and indicators	Successful
4	LLM Analysis	The system sends the results of the analysis to the LLM	URL phishing	The system displays interpretations and recommendations	Successful
5	Hybrid Detection	The system combines Rule-Based and LLM results	URLs analyzed	The system produces the final result	Successful
6	Database Storage	The results of the analysis are saved to the database	Data from the analysis	Data stored in analysis_history table	Successful
7	Analysis History	The user opens the history page	Stored data	The system displays a list of analysis history	Successful
8	Export PDF	The user presses the Download PDF button	Data from the analysis	PDF file successfully created and downloaded	Successful
9	Docker Backend	Frontend accessing backend API	URL analysis request	The backend responds without errors	Successful

10	Results Display	The system displays the results of the analysis	Valid URL	Phishing information is displayed in full	Successful

1. **URL Input**

   The test was conducted by entering a valid URL through the system interface. The test results showed that the system successfully accepted the URL input and forwarded it to the analysis process without any errors. Thus, the URL input feature functioned as designed.

2. **URL Validation**

   The test was conducted by entering an empty URL or an invalid URL format. The system successfully displayed an error message and prevented the analysis process from continuing. This result indicates that the input validation mechanism was functioning properly.

3. **Rule-Based Analysis**

   The test was conducted using URLs containing phishing indicators such as the use of the words "login," "verify," or suspicious domains. The system successfully calculated a risk score based on predefined rules and displayed analysis results based on the tested URL.

4. **LLM Analysis**

   The test was conducted by sending the URL analysis results to a Large Language Model (LLM). The system successfully generated additional interpretations in the form of risk explanations and security recommendations based on the characteristics of the analyzed URL.

5. **Hybrid Detection**

   The test was conducted by combining the results of Rule-Based Detection and LLM Analysis. The system successfully produced a final decision in the form of a Safe, Low Risk, Warning, or Phishing classification according to the designed hybrid logic.

6. **Database Storage**

   The test was conducted after the analysis process was completed. The results showed that the URL data, analysis score, LLM results, and final decision were successfully saved to the database table without data loss.

7. **Analysis History**

The test was conducted by opening the analysis history menu after some data was saved to the database. The system successfully displayed all the stored analysis data in a complete and sorted manner.

8. **Export PDF**

   The test was conducted by clicking the Export PDF button in the analysis results. The system successfully generated and downloaded a PDF document containing complete phishing analysis information in the designed report format.

9. **Docker Backend**

   The test was conducted by running the frontend, backend, and database through Docker. The test results showed that all services communicated effectively with each other, and no connection issues were found during the analysis process.

10. **Results Display**

    The test was conducted after the analysis process was completed. The system successfully displayed information such as the analyzed URL, Rule-Based score, LLM interpretation results, risk level, and final decision in a clear and easy-to-understand manner.

Tabel 3.8 Functionality Testing

No	Features	Status
1	URL Input	Successful
2	Rule-Based Detection	Successful
3	LLM Analysis	Successful
4	Hybrid Decision	Successful
5	Database Storage	Successful
6	Analysis History	Successful
7	PDF Export	Successful
8	Docker Connectivity	Successful

Description :

1. Succeful : The function operates as specified
2. Unsuccessful : The function is not working as specified

Based on the results of the functional testing conducted, all key features of the phishing detection system which combines rule-based and Large Language Model (LLM) technologies operate as intended. No errors were found that hindered the analysis process, data storage, or

report generation. Therefore, the system is deemed to have met the functional requirements established during the design phase and is ready for use in identifying phishing sites.

3.8 Dataset Preparation

The dataset used to evaluate the performance of the proposed phishing detection system consists of 100 URL samples. These samples serve as the ground truth for the evaluation process using a confusion matrix. The entire dataset was compiled from the Malicious URL dataset sourced from Kaggle.

As presented in Table 3.9, the original dataset contains four distinct URL classifications: benign (71 samples), defacement (21 samples), phishing (7 samples), and malware (1 sample). Because the proposed hybrid system uses four risk levels, namely Safe, Low Risk, Not Recommended, and Phishing, while the original dataset uses different threat labels, a label normalization process is performed before the evaluation.

The label normalization is applied to ensure consistency between the original dataset labels and the classification categories generated by the proposed system. The original benign label is mapped to the Safe category because it represents URLs without identified malicious characteristics. The defacement and malware labels are mapped to the Not Recommended category because both represent malicious activities that should not be accessed, but are not specifically classified as phishing by the proposed system. Meanwhile, the original phishing label is mapped directly to the Phishing category because it represents the specific threat that is the primary focus of this research.

Furthermore, it is important to acknowledge the class imbalance in this sample set. The dataset contains a large proportion of non-phishing URLs compared with actual phishing URLs, with 93 samples belonging to benign, defacement, and malware categories and only 7 samples belonging to the phishing category. This distribution results in a limited number of positive phishing samples for evaluating the system's ability to detect phishing threats. Therefore, the performance results should be interpreted with consideration of the class distribution within the dataset.

Tabel 3.9 URL Dataset for System Evaluation

No	URL	Type
1	br-icloud.com.br	phishing
2	mp3raid.com/music/krizz_kaliko.html	benign
3	bopsecrets.org/rexroth/cr/1.htm	benign
4	http://www.garage-pirenne.be/index.php?option=com_content&view=article&id=70&vsig70_0=15	defacement
5	http://adventure-nicaragua.net/index.php?option=com_mailto&tmpl=component&link=aHR0cDovL2FkdmVudHVyZS1uaWNhcmFndWEubmV0L2luZGV4LnBocD9vcHRpb249Y29tX2NvbnRlbnQmdmlldz1hcnRpY2xlJmlkPTQ3OmFib3V0JmNhdGlkPTM2OmRlbW8tYXJ0aWNsZXMmSXRlbWlkPTU0	defacement
6	http://buzzfil.net/m/show-art/ils-etaient-loin-de-s-imaginer-que-le-hibou-allait-faire-ceci-quand-ils-filmaient-2.html	benign
7	espn.go.com/nba/player/_/id/3457/brandon-rush	benign
8	yourbittorrent.com/?q=anthony-hamilton-soulife	benign
9	http://www.pashminaonline.com/pure-pashminas	defacement
10	allmusic.com/album/crazy-from-the-heat-r16990	benign
11	corporationwiki.com/Ohio/Columbus/frank-s-benson-P3333917.aspx	benign
12	http://www.ikenmijnkunst.nl/index.php/exposities/exposities-2006	defacement
13	myspace.com/video/vid/30602581	benign
14	http://www.lebensmittel-ueberwachung.de/index.php/aktuelles.1	defacement
15	http://www.szabadmunkaero.hu/cimoldal.html?start=12	defacement
16	http://larcadelcarnevale.com/catalogo/palloncini	defacement
17	quickfacts.census.gov/qfd/maps/iowa_map.html	benign
18	nugget.ca/ArticleDisplay.aspx?archive=true&e=1160966	benign
19	uk.linkedin.com/pub/steve-rubenstein/8/718/755	benign
20	http://www.vnic.co/khach-hang.html	defacement
21	baseball-reference.com/players/h/harrige01.shtml	benign
22	signin.eby.de.zukruygxctzmmqi.civpro.co.za	phishing
23	192.com/atoz/people/oakley/patrick/	benign
24	nytimes.com/1998/03/29/style/cuttings-oh-that-brazen-raucous-glorious-hibiscus.html	benign
25	escholarship.org/uc/item/5xt4952c	benign

26	songfacts.com/detail.php?id=13410	benign
27	casamanana.org/education/blba/	benign
28	http://hollywoodlife.com/2014/05/01/rihanna-iheartradio-music-awards-dress-2014-pics/	benign
29	http://www.marketingbyinternet.com/mo/e56508df639f6ce7d55c81ee3fcd5ba8/	phishing
30	en.wikipedia.org/wiki/North_Dakota	benign
31	soaps.sheknows.com/daysofourlives/news/id/20259/Days_Of_Our_Lives_Casts_Ruta_Lee/	benign
32	perfectpeople.net/celebrity-star/6920/gidget-gein.htm	benign
33	media.caltech.edu/press_releases/13186	benign
34	wikiobits.com/Obits/TonyProudfoot	benign
35	vanderbilt.rivals.com/viewcoach.asp?coach=2079&sport=1&year=2011	benign
36	http://peluqueriadeautor.com/index.php?option=com_virtuemart&page=shop.browse&category_id=31&Itemid=70	defacement
37	movies.yahoo.com/shop?d=hv&cf=info&id=1800340831	benign
38	cyndislist.com/us/pa/counties	benign
39	http://www.824555.com/app/member/SportOption.php?uid=guest&langx=gb	malware
40	http://www.raci.it/component/user/reset.html	defacement
41	https://docs.google.com/spreadsheet/viewform?formkey=dGg2Z1lCUHlSdjllTVNRUW50TFIzSkE6MQ	phishing
42	psychology.wikia.com/wiki/Phonemes	benign
43	info.centriq.com/content/fivereasons	benign
44	articles.baltimoresun.com/1991-06-11/sports/1991162162_1_james-koehler-texas-rangers-terrell-lowery	benign
45	infinitysw.com/	benign
46	strawberrycreekgardens.com/	benign
47	http://earth-sure.com/industrial-tech/everlast-impact-resistant-dumpster-enclosure	defacement
48	spoke.com/dir/p/desantis/nick	benign
49	wordsmith.org/words/pignus.html	benign
50	http://interpark.com/displaycorner/FreeMarket.do?_method=itemCateList&free1=pc&free2=007001008003&free3=item&disp_no=007001008003	benign
51	http://www.newtec.ac.uk/courses/health-and-social-care/level-2-diploma-in-health-a-social-care-.html	defacement

52	http://www.kingsmillshotel.com/spring/mothers-day	defacement
53	almanac.logos.com/Denominations	benign
54	gaming.wikia.com/wiki/Backbreaker_(video_game)	benign
55	ceu.hu/	benign
56	registry.weddingchannel.com/coupledir/20108/C/R314763282/JARED_CLOUGH_AND_TAJAI_RITCHIE.htm	benign
57	linkedin.com/pub/george-cleveland/7/80a/599	benign
58	sunshinemills.com/	benign
59	agoda.com/north_america/canada/montreal_qc/attractions/hotels_near_mcgill_university_centre_for_continuing_education.html	benign
60	ctv.ca/generic/WebSpecials/sponsorship_scandal/	benign
61	http://www.approvi.com.br/ck.htm	defacement
62	http://www.shaborooz.ir/index.php?view=article&catid=7:1390-01-27-09-09-55&id=23:1390-04-12-08-07-33&tmpl=component&print=1&layout=default&page=&option=com_content&Itemid=2	defacement
63	veoox.com/tags/game.html	benign
64	chuckwalkertrio.blogspot.com/	benign
65	aerosolesshoes.org/	benign
66	http://www.juventudelirica.com.br/index.html	defacement
67	http://www.myenrg.com/southwest/9-texas	defacement
68	recipelink.com/msgbrd/board_14/2007/DEC/29094.html	benign
69	http://www.musimagen.com/lista_socios.php?letra=\303%91	defacement
70	ideas.repec.org/f/pma235.html	benign
71	peopleaz.org/firstname/Fowlkes/6	benign
72	orensamtraktrip.wordpress.com/	benign
73	retajconsultancy.com	phishing
74	alexa.com/whatshot?q=james+arness	benign
75	jpopasia.com/group/arashi/discography/	benign
76	kitsapsun.com/photos/2011/feb/25/177999/	benign
77	local.yahoo.com/info-42862457-oakland-international-airport-oakland	benign
78	tripadvisor.com/Restaurant_Review-g680765-d1768647-Reviews-Yacha_Yacha-Miyakojima_Okinawa_Prefecture_Kyushu_Okinawa.html	benign
79	articles.elitefts.com/articles/training-articles/five-great-exercises-for-building-a-dominant-defensive-lineman/	benign

80	broadwaystars.com/classic/?source=www.playbill.com	benign
81	http://www.pn-wuppertal.de/links/2-linkseite/5-httpwwwkrebshilfede	defacement
82	forums.networkinfrastructure.info/motorola-wireless-lan-switches/	benign
83	youtube.com/watch?v=mwpoPQ1SPJU	benign
84	mccall-idchamber.org/	benign
85	chacha.com/question/who-said-the-quote-%27say-what-you-mean%2C-mean-what-you-say-and-put-a-beat-to-it%27	benign
86	friars.com/sports/m-baskbl/archive/prov-m-baskbl-2003.html	benign
87	http://asociacionkaribu.org/index.php/servicios/centro-jambo	defacement
88	http://www.martin-busker.de/administrator/help/en-GB/css/Facture/c4d12146ebce8e1684d3542308399779/8fa39dab95edb1b676b638a672278eae/particuliers-45636.php	phishing
89	pipl.com/directory/name/Whissell/180	benign
90	spokeo.com/John+Collet	benign
91	http://www.parafiapiaski.pl/index.php?option=com_user&view=remind	defacement
92	http://fanpage.gr/family/gineka/%ce%b1%cf%85%cf%84%ce%bf%ce%af-%ce%b5%ce%af%ce%bd%ce%b1%ce%b9-%ce%bf%ce%b9-13-%cf%89%cf%81%ce%b1%ce%b9%cf%8c%cf%84%ce%b5%cf%81%ce%bf%ce%b9-%ce%ad%ce%bb%ce%bb%ce%b7%ce%bd%ce%b5%cf%82-%ce%b1%ce%bd-2/	benign
93	faqs.org/copyright/handbook-for-the-individual-or-group-culture-free/	benign
94	duckduckgo.com/1/c/Roman_Catholic_cathedrals_in_Canada	benign
95	alexpay2.beget.tech	phishing
96	http://worldoftanks.ru/ru/content/guide/payments_instruction/mobile-payments-rostelekom-ural-utel/	benign
97	http://www.artedesignsas.it/catalogo.html?page=shop.browse&category_id=14	defacement
98	en.wikipedia.org/wiki/Lloyd_Bacon	benign
99	http://serverfault.com/questions/682832/squid-transparent-https-certificate-error-ssl-error-bad-cert-domain	benign
100	http://www.viamanaus.com.br/index.php?option=com_wrapper&view=wrapper&Itemid=297	defacement

Source : Kaggle Phishing URLs Dataset

For system predictions, the Safe and Low Risk outputs are normalized into the Safe evaluation category. This is because both categories represent URLs that have not exceeded the system's direct threat threshold. The Not Recommended output remains categorized as Not Recommended, while the Phishing output remains categorized as Phishing. Therefore, the normalized evaluation uses three final categories: Safe, Not Recommended, and Phishing. This normalization is applied consistently before the construction of the confusion matrix and the calculation of Accuracy, Precision, Recall, and F1-Score.The complete list of URLs used in this testing is presented in Table 3.9.

The resulting label mapping is presented in the Tabel :

Tabel 3.10 Label Normalization from Original Dataset to System Classification

Original Dataset Label	**Normalized Ground Truth**
Benign	Safe
Defacement	Not Recommended
Malware	Not Recommended
Phishing	Phishing

Table 3.10 illustrates the process of normalizing the original dataset labels into the ground-truth classifications used for system evaluation. The initial dataset comprised four categories Benign, Defacement, Malware, and Phishing whereas the system employs a different risk classification scheme. Consequently, the "Benign" label was normalized to "Safe," as it represents URLs not identified as threats. The "Defacement" and "Malware" labels were merged into "Not Recommended," as both represent categories of malicious URLs that are not specifically classified as phishing by the system. Meanwhile, the "Phishing" label remained "Phishing," as it corresponds directly to the system's phishing category.

Thus, following the normalization process, the four original dataset labels were transformed into three ground-truth classes: Safe, Not Recommended, and Phishing. This normalization was performed prior to constructing the confusion matrix and calculating evaluation metrics to ensure that all test data carried labels consistent with the system's classification scheme.

For the system prediction, the normalization is defined as follows :

Tabel 3.11 System Prediction Normalization for Performance Evaluation

System Output	Normalized Prediction
Safe	Safe
Low Risk	Safe
Not Recommended	Not Recommended
Phishing	Phishing

Table 3.11 illustrates the normalization process applied to the system's prediction output prior to performance evaluation. During the analysis phase, the system generates four risk levels: Safe, Low Risk, Not Recommended, and Phishing. To align the system output with the three ground-truth classes established in Table 3.10, the "Safe" and "Low Risk" categories are merged into a single "Safe" category. This consolidation is performed because both categories represent URLs that have not reached the threat levels classified as "Not Recommended" or "Phishing."

Meanwhile, the "Not Recommended" and "Phishing" outputs remain unchanged. Consequently, all system prediction results used in the final testing fall into the same three classes as the ground truth: Safe, Not Recommended, and Phishing.

3.9 Peformance Evaluation

To evaluate the effectiveness of the proposed hybrid phishing detection system, several classification performance metrics were employed, namely Accuracy, Precision, Recall, and F1-Score. These metrics were calculated based on the Confusion Matrix, which consists of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The Confusion Matrix represents the relationship between the actual labels and the predicted labels generated by the system.

1. True Positive (TP) refers to phishing URLs correctly classified as phishing.
2. True Negative (TN) refers to legitimate URLs correctly classified as safe.
3. False Positive (FP) refers to legitimate URLs incorrectly classified as phishing.
4. False Negative (FN) refers to phishing URLs incorrectly classified as safe.

The performance metrics are calculated using the following equations.

1. Accuracy

Accuracy measures the overall ability of the system to correctly classify both phishing and legitimate URLs.

$$Accutacy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP (True Positive) = Phishing URLs that were successfully detected as phishing.
- TN (True Negative) = Safe URLs that were successfully detected as safe.
- FP (False Positive) = Safe URLs that were incorrectly detected as phishing.
- FN (False Negative) = Phishing URLs that were incorrectly detected as safe.

A higher Accuracy value indicates that the system can correctly classify a larger proportion of URLs.

2. Precision

Precision measures the proportion of URLs classified as phishing that are actually phishing. This metric indicates how accurate the system is when predicting a URL as phishing.

$$Precision = \frac{TP}{TP + FP}$$

where:

- TP (True Positive) = Number of phishing URLs correctly classified as phishing.
- FP (False Positive) = Number of legitimate URLs incorrectly classified as phishing.

A high Precision value indicates that the system produces fewer false phishing alarms.

3. Recall

Recall measures the ability of the system to correctly identify phishing URLs from all actual phishing URLs. This metric is important in phishing detection because it indicates how many phishing attacks can be successfully detected by the system.

$$Recall = \frac{TP}{TP + FN}$$

where:

- TP (True Positive) = Number of phishing URLs correctly classified as phishing.

- FN (False Negative) = Number of phishing URLs incorrectly classified as safe.

A high Recall value indicates that the system can successfully identify most phishing URLs and minimize undetected attacks.

4. F1-Score

F1-Score is the harmonic mean of Precision and Recall. This metric provides a balanced measure of the system's performance by considering both false positives and false negatives. F1-Score is particularly useful when the dataset is imbalanced or when both Precision and Recall are equally important.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where:

- Precision = Proportion of URLs classified as phishing that are actually phishing.
- Recall = Proportion of actual phishing URLs correctly identified by the system.

A higher F1-Score indicates a better balance between phishing detection capability and classification accuracy.

CHAPTER IV
DISCUSSION AND RESULTS

4.1 How The System works

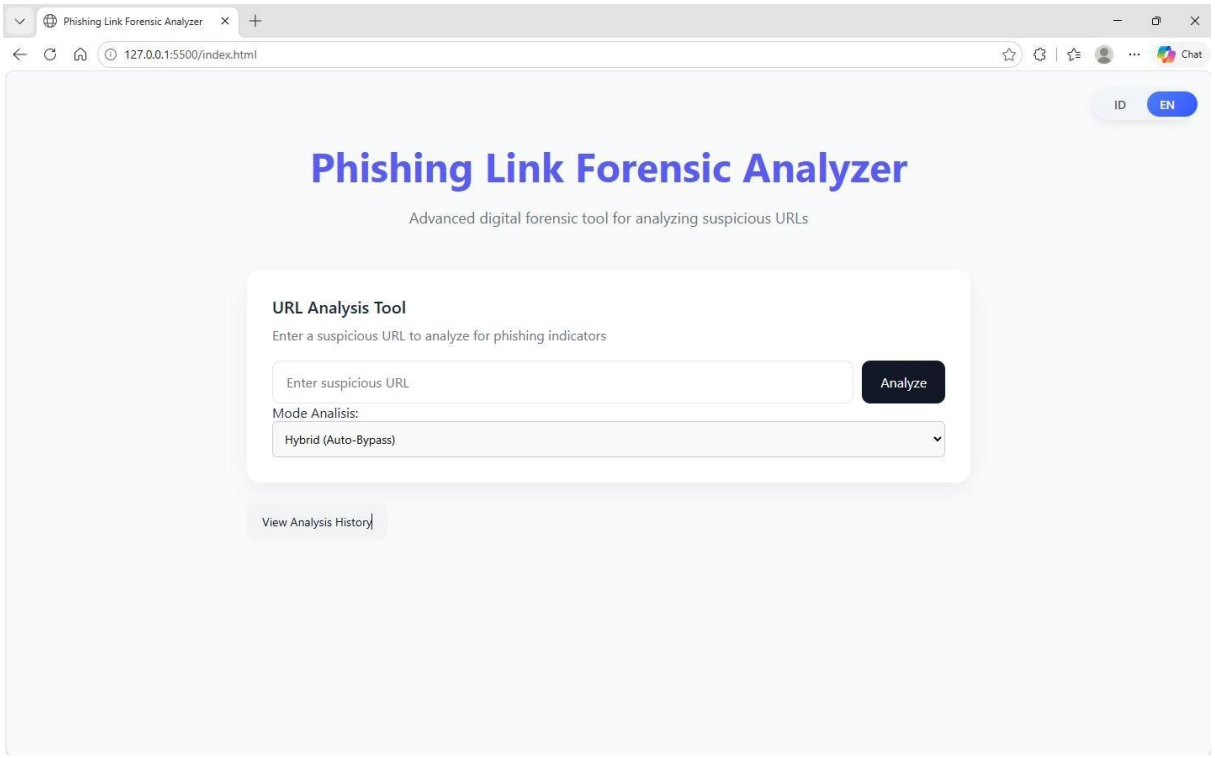


Figure 4.1 Home Page

Figure 4.1 illustrates the main interface of the web-based phishing detection system developed in this study. The analysis process initiates when a user accesses the platform via a web browser and inputs a target URL into the designated text field. Upon entering the URL, the user clicks the "Analyze" button to execute the digital forensic process. Additionally, the interface incorporates a seamless language translation module, indicated by the ID/EN toggle in the top right corner, ensuring the tool is accessible to a broader user base by supporting dual-language localized functionality.

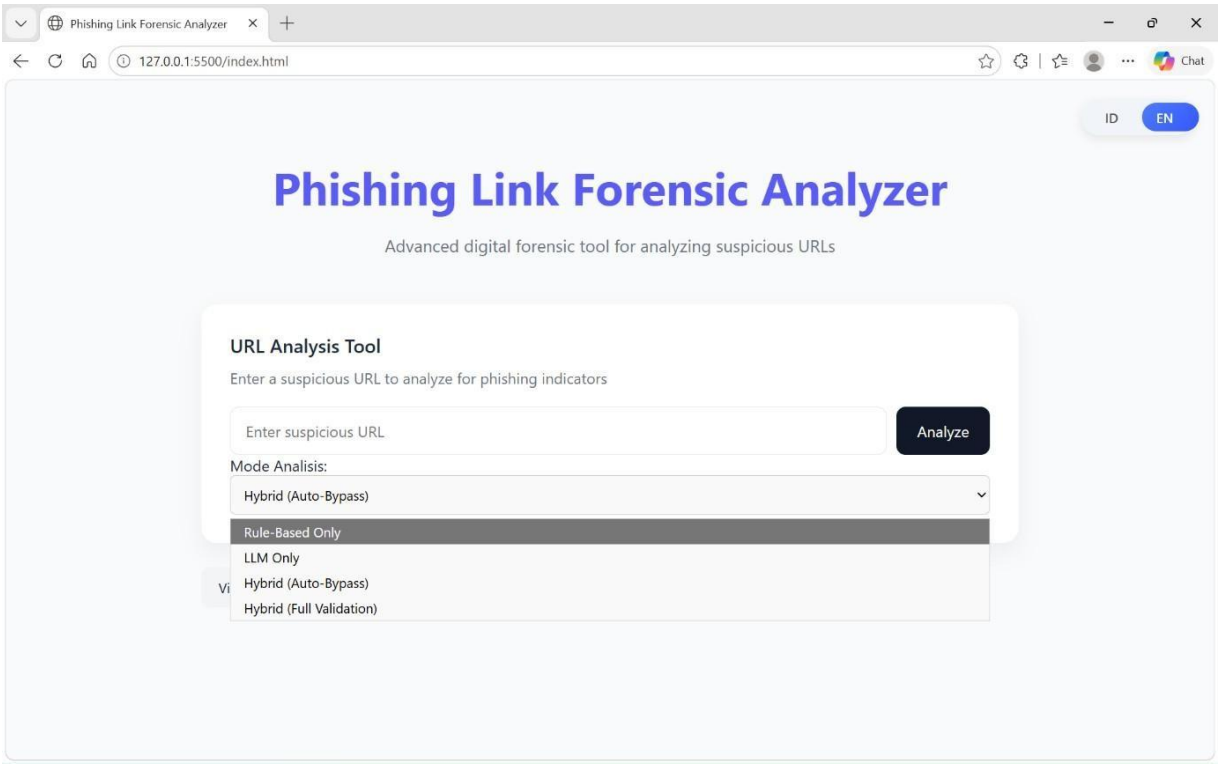


Figure 4.2 Analysis Mode Selection

Figure 4.2 depicts the "Mode Analisis" dropdown menu, which governs the operational architecture of the system during the evaluation process. As illustrated in the expanded menu, the system provides users with four distinct analytical configurations to accommodate varying forensic requirements. The first option, "Rule-Based Only," relies exclusively on deterministic, static heuristics to evaluate the URL structure, bypassing the Large Language Model (LLM) entirely to offer rapid detection based on predefined thresholds. Conversely, the "LLM Only" mode routes the analysis directly to the LLM module for purely contextual and interpretive evaluation without static interventions. For a more integrated approach, the "Hybrid (Auto-Bypass)" mode employs a resource-efficient architecture where the URL is initially evaluated by the rule-based engine, and the LLM is subsequently invoked only if the extracted indicators exhibit ambiguous patterns. Alternatively, the "Hybrid (Full Validation)" option offers a comprehensive analytical approach where every submitted URL undergoes rigorous evaluation by both the deterministic engine and the LLM, ensuring maximum analytical depth regardless of the initial static score. Ultimately, this dropdown mechanism provides users with a streamlined and flexible interface to explicitly define their preferred analytical approach prior to initiating the digital forensic scan, seamlessly bridging deterministic rules with advanced contextual AI.

.

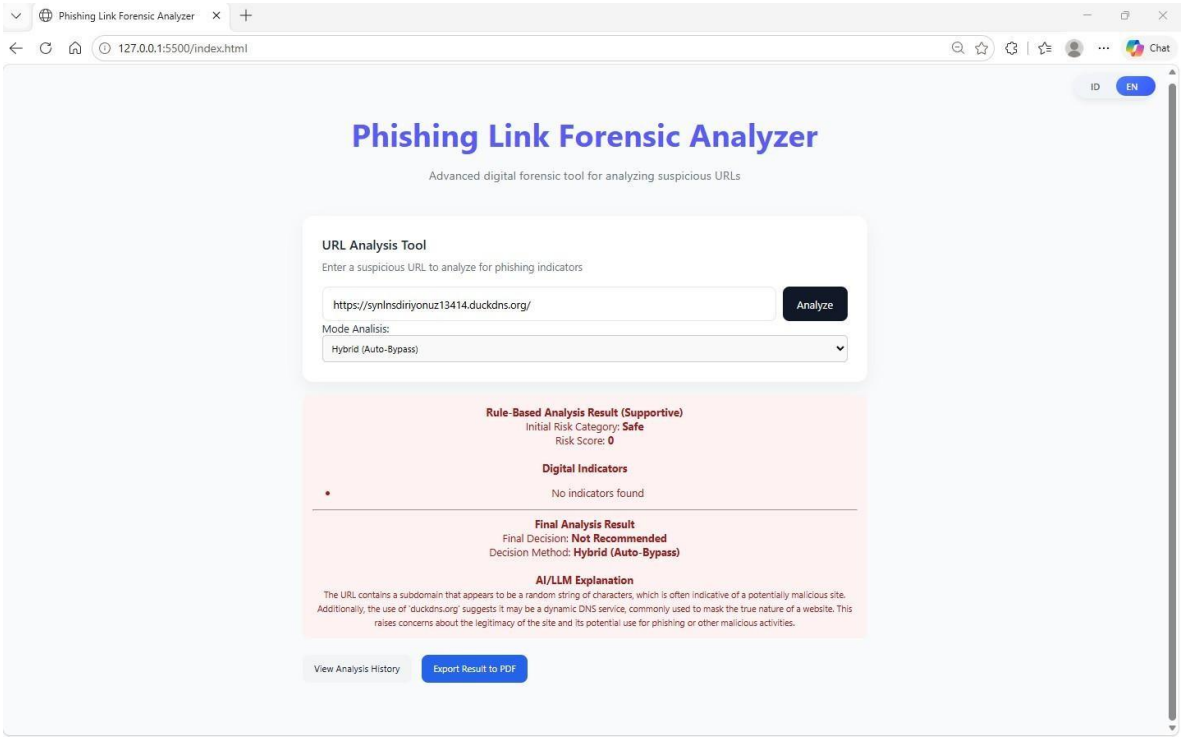


Figure 4.3 URL Analysis Process and Results

Figure 4.3 illustrates the execution of the URL analysis process and the comprehensive presentation of its diagnostic results within the web interface. Upon submission, the frontend transmits the target URL to the backend server for an initial deterministic evaluation using the rule-based engine. During this phase, the system scans the URL string for predefined digital phishing artifacts, such as the presence of IP addresses acting as domains, "@" symbols, abnormal lengths, sensitive keywords, excessive subdomains, URL shorteners, and domain hyphens. This initial scan generates a baseline risk score and a supportive risk category. Subsequently, as dictated by the selected "Hybrid (Auto-Bypass)" mode, the analysis is forwarded to the Large Language Model (LLM) module for rigorous contextual evaluation. As prominently demonstrated in the figure, the LLM is capable of identifying complex evasion patterns such as the use of dynamic DNS services (e.g., duckdns.org) and randomized subdomain strings that successfully bypassed the static rules. Ultimately, the interface presents a transparent forensic summary that strictly demarcates the supportive rule-based findings from the final diagnostic decision, accompanied by a detailed Ai generated explanation. Additionally, the system provides integrated functionalities at the bottom of the interface, allowing users to review previous analysis logs or export the detailed diagnostic results into a

dynamically generated PDF report for further documentation and reporting purposes.

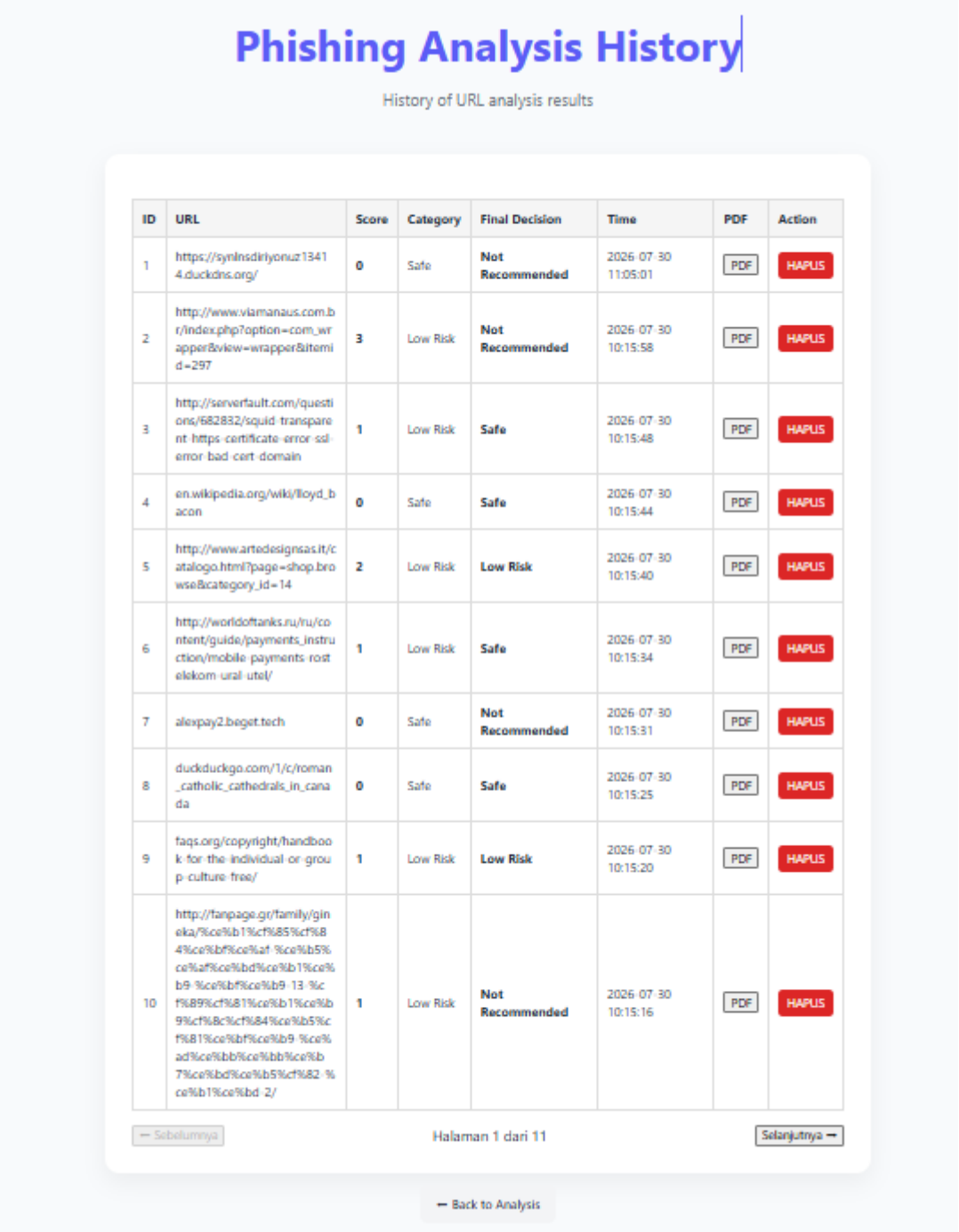


Figure 4.4 Storage of Analysis History Results

Figure 4.4 Storage of Analysis Result History, all analysis results will be stored in the database for the purpose of analysis history and PDF report creation. The system also provides a View Analysis History feature to view the history of analysis results that have been performed previously. With this workflow, the system is able to assist users in identifying potential phishing early based on automatic and systematic analysis of URL structures.

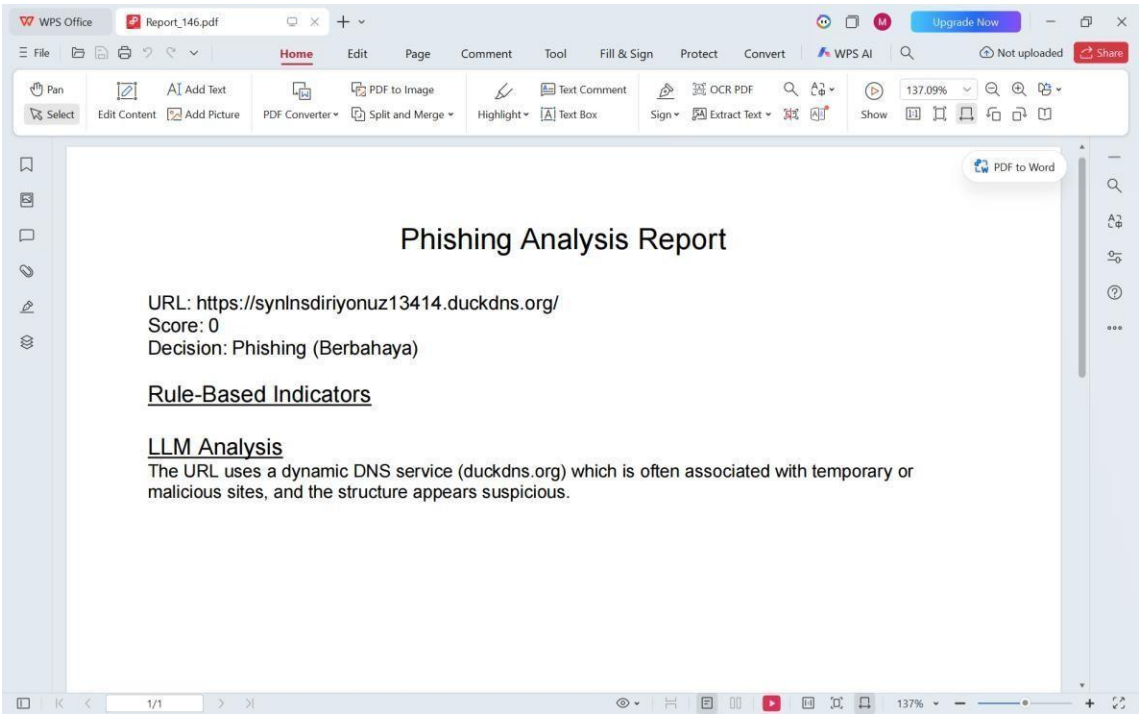


Figure 4.5 PDF Report Genertor

Figure 4.5 shows the PDF Report Generator feature that functions to automatically generate phishing analysis reports in PDF format. This feature allows users to save and document the results of URL checks that have been performed by the system. The resulting report contains important information, such as the analyzed URL, the detection score, the final decision of the system, the indicators detected by the rule-based method, and the results of interpretation and explanation from the Large Language Model (LLM).

In the example shown, the system analyzes the https://synlnsdiriyonuz13414.duckdns.org/ URL and generates a Phishing (Dangerous) decision. In addition to displaying the results of the classification, the report also includes an explanation from the LLM explaining the reasons behind the decision. In these cases, the LLM identifies that the domain is using a Dynamic DNS service (DuckDNS) that is often associated with temporary sites or suspicious activity, thereby increasing the risk level of those URLs.

The PDF Report Generator feature provides benefits in the process of documenting and reporting analysis results, especially in digital forensic investigation activities. With the existence of reports in PDF format, the results of the analysis can be saved, printed, shared, or used as supporting documentation in the evaluation and research process. In addition, the presentation of results in the form of structured reports helps users understand the detected indicators and the reasons underlying the system's decisions more clearly and systematically.

4.2 System Implementation

The phishing detection system is implemented web-based using both frontend and backend architectures. The frontend is used as a user interface to enter URLs, display analysis results, and download detection results reports in PDF format. The backend is developed using Node.js and Express.js that serve as data processing centers and system controllers. In its implementation, the system uses rule-based methods to detect known phishing indicators, such as the use of IP addresses, the "@" symbol, abnormal URL lengths, sensitive keywords, the use of URL shorteners, the number of subdomains, and the use of hyphens on domains.

The results of the initial analysis are then forwarded to the LLM module using the GPT-4o-mini model for further analysis of the URL pattern more thoroughly. The system is also equipped with a database to store analysis data, including URLs, rule-based scores, LLM analysis results, final classification, and timestamps. In addition, the system provides a feature for generating analysis reports in PDF format to support documentation of phishing detection results. At the implementation stage, the backend and database are run using Docker to maintain the consistency of the system environment, simplify the deployment process, and improve the stability of application services.

4.3 System Test Results

System testing is done through several test scenarios that include URL input processes, rule-based analysis, integration with LLMs, data storage, and PDF report generation. The test was conducted using several examples of URLs consisting of safe URLs and URLs that are indicated by phishing. The test results show that the system is capable of receiving and processing URLs well through the web interface that has been provided. In the rule-based analysis stage, the system successfully detected several phishing indicators such as the use of IP addresses as domains, the use of the "@" symbol, the use of unusual URL lengths, the use of sensitive keywords such as login, verify, update, and account, the use of URL shorteners, the number of subdomains excessive, and the use of hyphens on domains.

After the initial analysis process is completed, the results are sent to the LLM module for further analysis. The LLM module successfully provides additional interpretation of URL patterns and produces the final classification in the form of Safe, Low Risk, Not

Recommended, and Phishing. In addition to producing the final classification, the system is also able to provide a more detailed explanation of the results of the analysis based on the relationship between the phishing indicators found. Data storage tests show that all analysis results are successfully saved to the database properly. Furthermore, the PDF report creation feature also runs according to the designed function and is capable of generating phishing analysis reports automatically.

4.3.1 Input Test URL

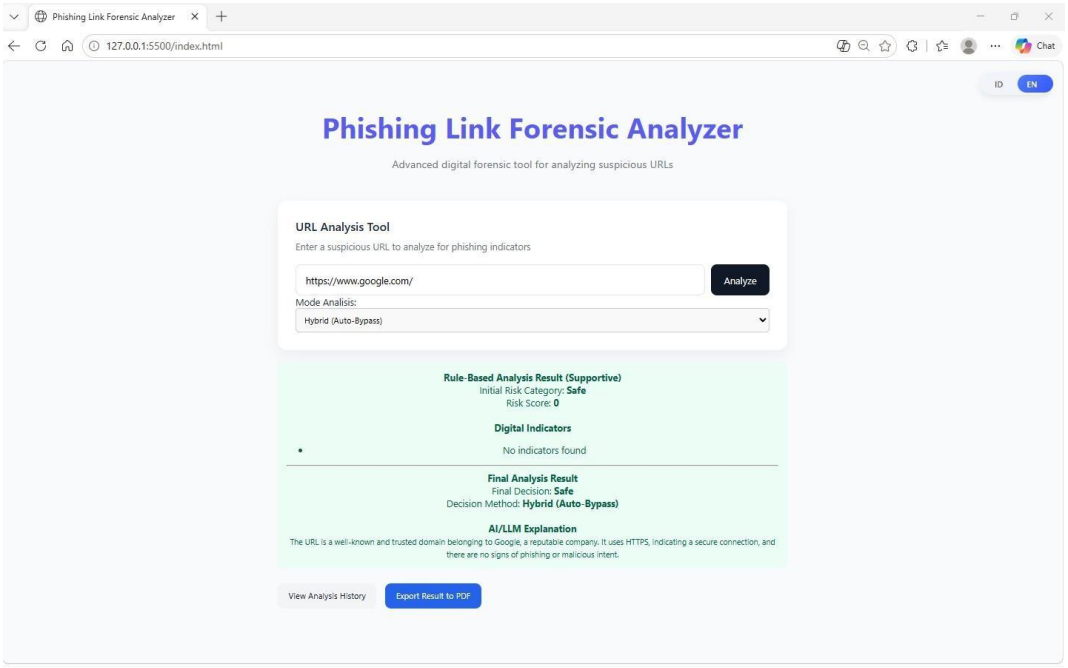


Figure 4.6 Analysis Result

Figure 4.6 illustrates the system's diagnostic output when a benign URL, specifically (https://www.google.com/), is submitted to the Phishing Link Forensic Analyzer. Operating under the "Hybrid (Auto-Bypass)" mode, the system effectively processes the input through its dual-layered architecture. The initial deterministic evaluation by the rule-based engine yields a risk score of 0 and classifies the URL within the "Safe" initial risk category, explicitly confirming the absence of any suspicious digital indicators. Concurrently, the LLM module evaluates the contextual properties of the input, accurately identifying it as a well-known, trusted domain belonging to a reputable corporate entity while highlighting the presence of a secure HTTPS connection. Synthesizing these evaluations, the system formulates a final decision of "Safe" via the Hybrid (Auto-Bypass) decision method. The visual feedback, denoted by the green-highlighted interface and the comprehensive AI generated explanation, demonstrates the system's capability to accurately validate legitimate web infrastructure while

providing transparent, user-friendly justifications to support its classification process.

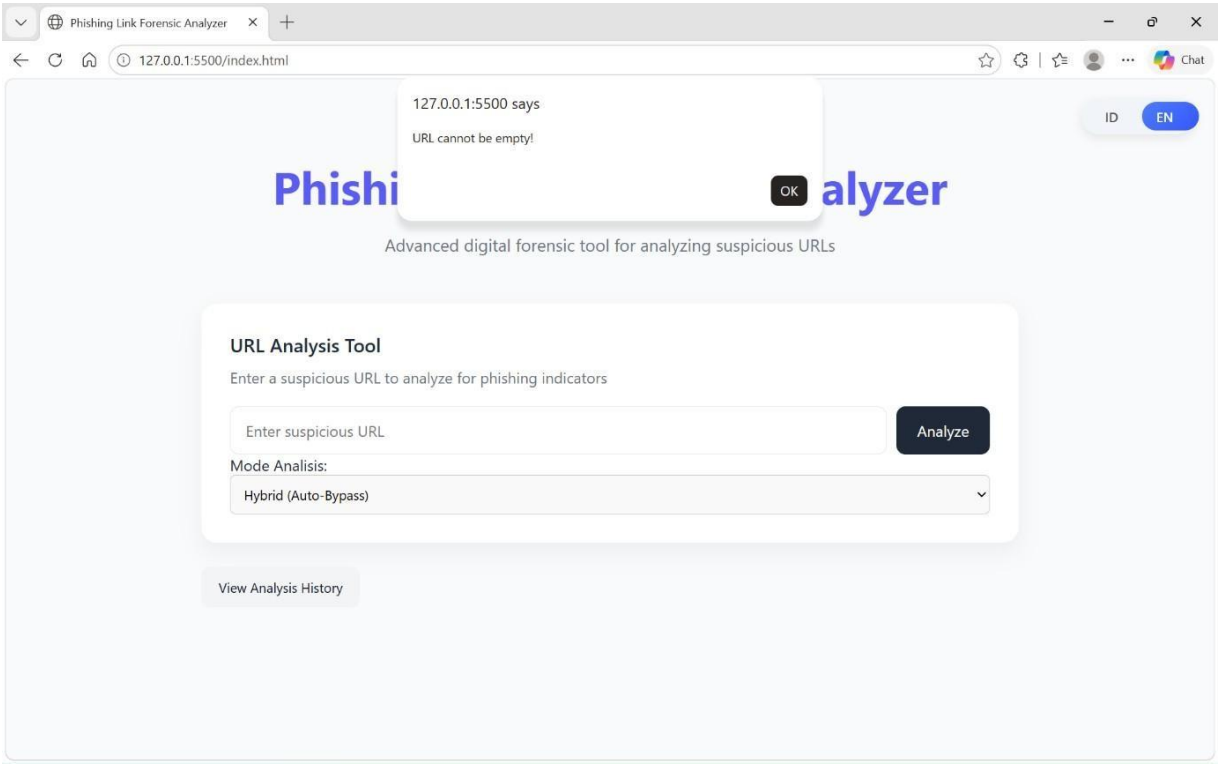


Figure 4.7 URL Validation

Figure 4.7 shows the result of testing the URL input validation feature on the Phishing Link Forensic Analyzer system. In this test, the Analyze button was pressed without entering any URL in the input field. The system successfully detected that the input field was empty and displayed a warning message stating "URL cannot be empty!". This validation mechanism prevents the system from processing invalid input and ensures that only valid URLs are analyzed. The successful appearance of the warning message indicates that the input validation feature functions correctly and is able to minimize user input errors before the phishing analysis process is executed.

4.3.2 Rule-Based Analysis

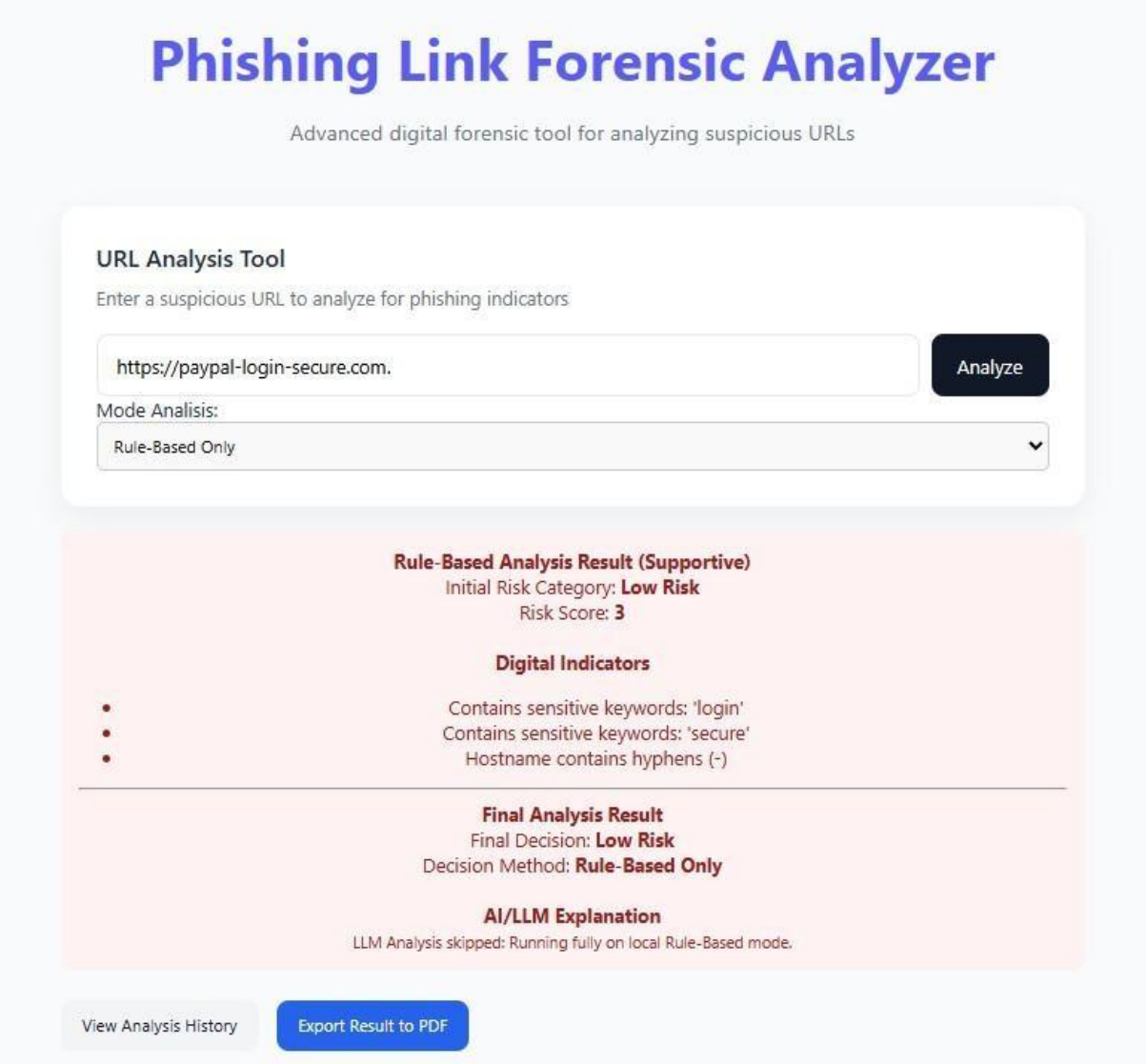


Figure 4.8 Rule-Based Analysis

Figure 4.8 illustrates the system's diagnostic output when evaluating a simulated phishing URL, (https://paypal-login-secure.com). Upon submission, the analysis is processed using the integrated detection framework. The rule-based engine evaluates the URL structure, assigning a risk score of 3 and successfully identifying several critical digital indicators, specifically the use of the sensitive words "login" and "secure," as well as the presence of a hyphen in the hostname, which is commonly used to mimic official domains. In accordance with the Smart LLM Filter (Auto-Bypass) mechanism, when the threat pattern derived from the static indicators is already deemed sufficiently clear, the system can efficiently conclude the assessment based on the heuristic results. Overall, these test results demonstrate the system's

capability to effectively capture structural anomalies and provide classifications that accurately align with identified risk levels.

4.3.3 LLM Analysis

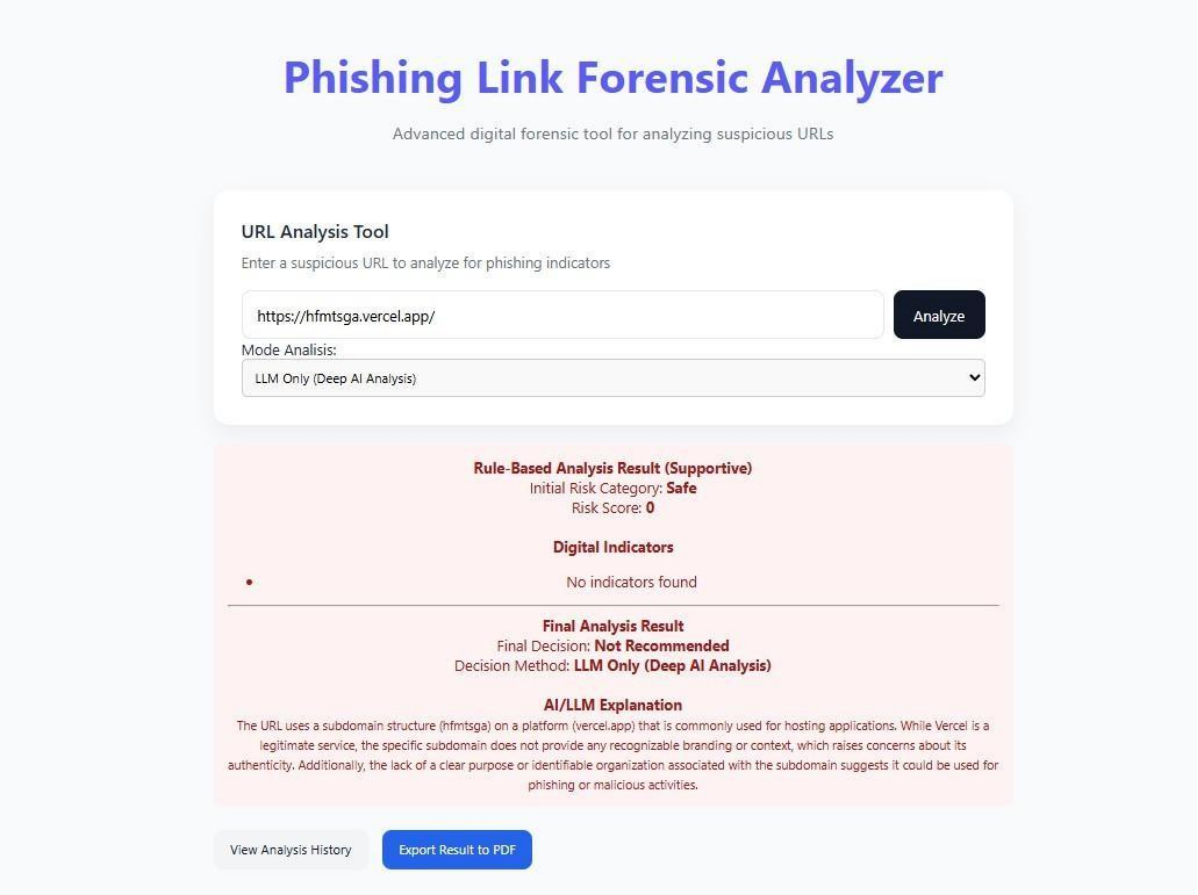


Figure 4. 9 LLM Analysis

Figure 4.9 illustrates the test results of the standalone LLM analysis feature when evaluating the URL (https://hfmthsga.vercel.app/) under the "LLM Only" operational mode. Although the deterministic rule-based engine yielded an initial risk score of 0 with no static indicators found, the input was routed exclusively to the Large Language Model (LLM) for deep contextual and interpretive evaluation. The LLM successfully identified potential security risks, noting that the subdomain hfmthsga consists of a random string lacking any recognizable brand affiliation or entity. Furthermore, the analysis highlighted that while vercel.app is a legitimate application deployment platform, such services can be exploited to host malicious content. Consequently, the system generated a final decision of "Not Recommended," demonstrating that the LLM module effectively captures contextual evasion tactics and provides sophisticated risk evaluations even when the underlying URL lacks explicit static indicators.

4.3.4 Hybrid (Auto-Bypass)

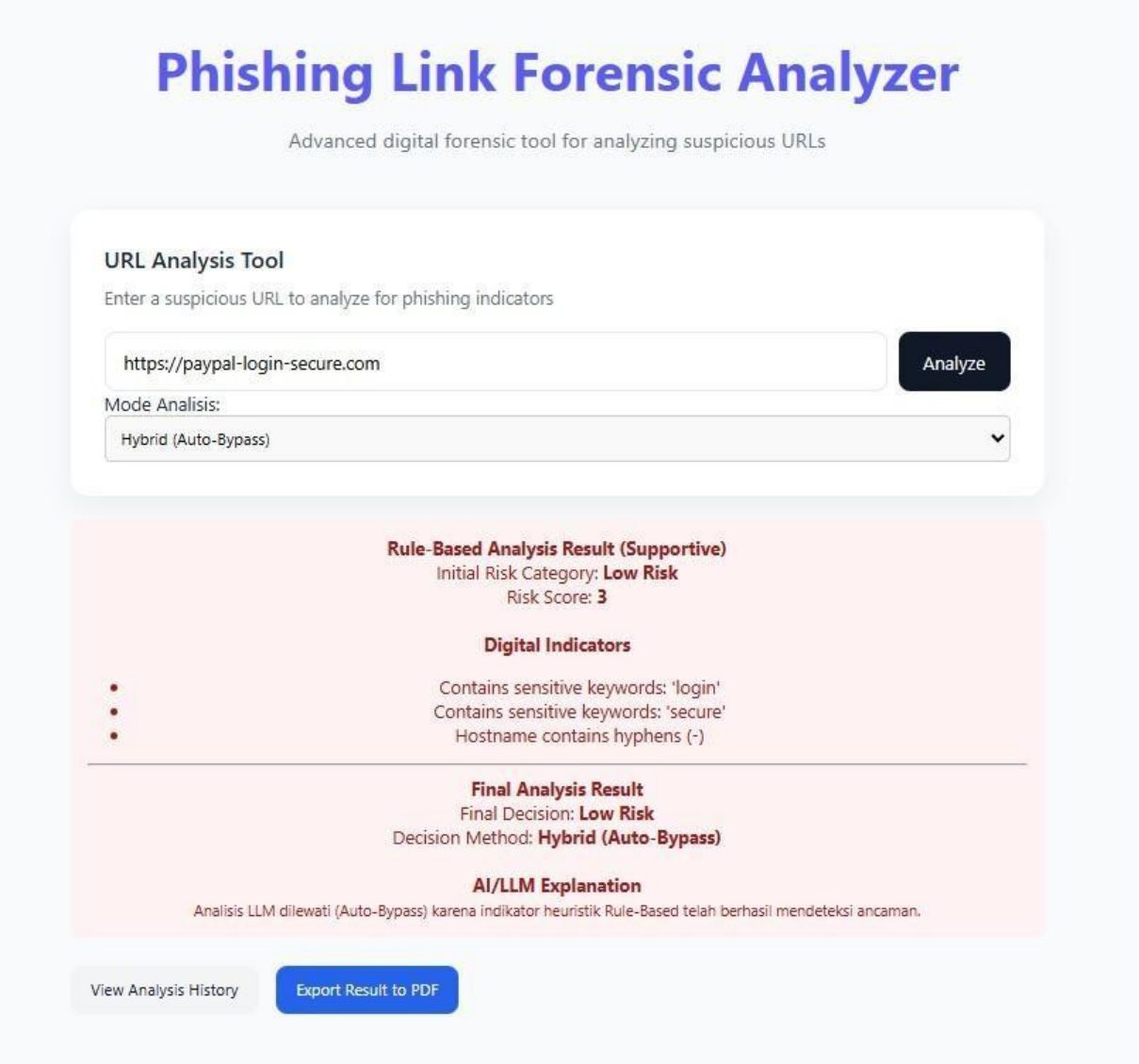


Figure 4.10 Hybrid (Auto-Bypass)

Figure 4.10 illustrates the test results of the Hybrid (Auto-Bypass) detection feature using the URL (https://paypal-login-secure.com). In this test, the system performed an initial analysis using the rule-based method, which yielded a risk score of 3 and categorized the input as "Low Risk" while identifying several specific digital indicators: the presence of the sensitive keywords "login" and "secure," as well as the use of hyphens in the hostname. Because the "Hybrid (Auto-Bypass)" mode is selected, the system evaluates whether the heuristic threat pattern is already sufficiently clear. As shown in the AI/LLM Explanation section of the interface ("Analisis LLM dilewati (Auto-Bypass) karena indikator heuristik Rule-Based telah berhasil mendeteksi ancaman"), the LLM analysis was bypassed because the rule-based indicators successfully captured the threat. Consequently, the final analysis result determined

a "Low Risk" decision via the Hybrid (Auto-Bypass) method, demonstrating how the auto-bypass mechanism optimizes performance by skipping redundant LLM evaluations when static rules provide conclusive indicators.

4.3.5 Hybrid (Full Validation)

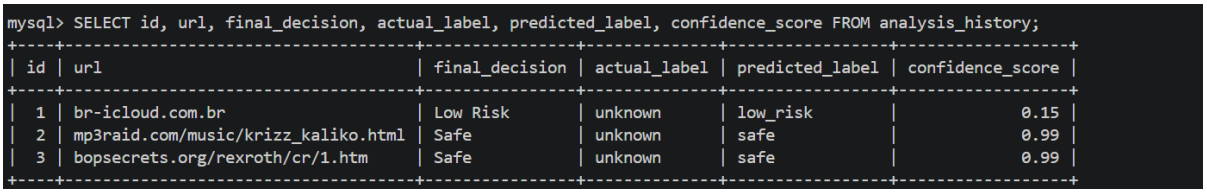


Figure 4.11 Hybrid (Full Validation)

Figure 4.11 illustrates the test results of the Hybrid (Full Validation) detection feature using the URL (https://paypal-login-secure.com). In this comprehensive evaluation mode, the system first executes an initial rule-based analysis, yielding a risk score of 3 and categorizing the input as "Low Risk" while identifying specific static indicators, namely the presence of the sensitive keywords "login" and "secure," as well as hyphens within the hostname. Unlike the auto-bypass configuration, the "Hybrid (Full Validation)" mode mandates that the system forwards the extracted findings to the Large Language Model (LLM) module for exhaustive contextual scrutiny regardless of the initial heuristic score. Consequently, the LLM generates a detailed

interpretive explanation, highlighting that the domain deceptively mimics the legitimate PayPal login page, exploits the term "secure" to foster false trust, and employs a structure designed to capture sensitive user credentials. Synthesizing both the deterministic indicators and the deep contextual reasoning, the system establishes a definitive final decision of "Phishing" via the Hybrid (Full Validation) decision method, demonstrating the architecture's capacity for thorough, dual-layered threat verification.

4.3.6 Database Storage



```
mysql> SELECT id, url, final_decision, actual_label, predicted_label, confidence_score FROM analysis_history;
+----+-------------------------------------+----------------+--------------+-----------------+------------------+
| id | url                                 | final_decision | actual_label | predicted_label | confidence_score |
+----+-------------------------------------+----------------+--------------+-----------------+------------------+
|  1 | br-icloud.com.br                    | Low Risk       | unknown      | low_risk        |             0.15 |
|  2 | mp3raid.com/music/krizz_kaliko.html | Safe           | unknown      | safe            |             0.99 |
|  3 | bopsecrets.org/rexroth/cr/1.htm     | Safe           | unknown      | safe            |             0.99 |
+----+-------------------------------------+----------------+--------------+-----------------+------------------+
```

Figure 4.12 Data Storage

Figure 4.12 Database Storage testing was conducted to ensure that the system can store URL analysis results in a MySQL database running in a Docker container. Once the analysis process is complete, the system automatically saves the analysis results to the `analysis_history` table. The stored data includes the analyzed URL, rule-based score, risk category, final decision, LLM analysis results, detected indicators, and analysis time. Based on the results of a database query, it is evident that the analysis results have been successfully stored and can be retrieved without any loss of information. Some of the stored data includes the URL https://paypal-login-secure.com, classified as Phishing (Dangerous) with the category VERIFIED BY LLM, and the URL https://hfmthsga.vercel.app, classified as Safe with the category LOW RISK. These results indicate that the data storage mechanism is functioning as designed and is capable of supporting the continuous documentation of analysis results.

4.3.7 Analysis History

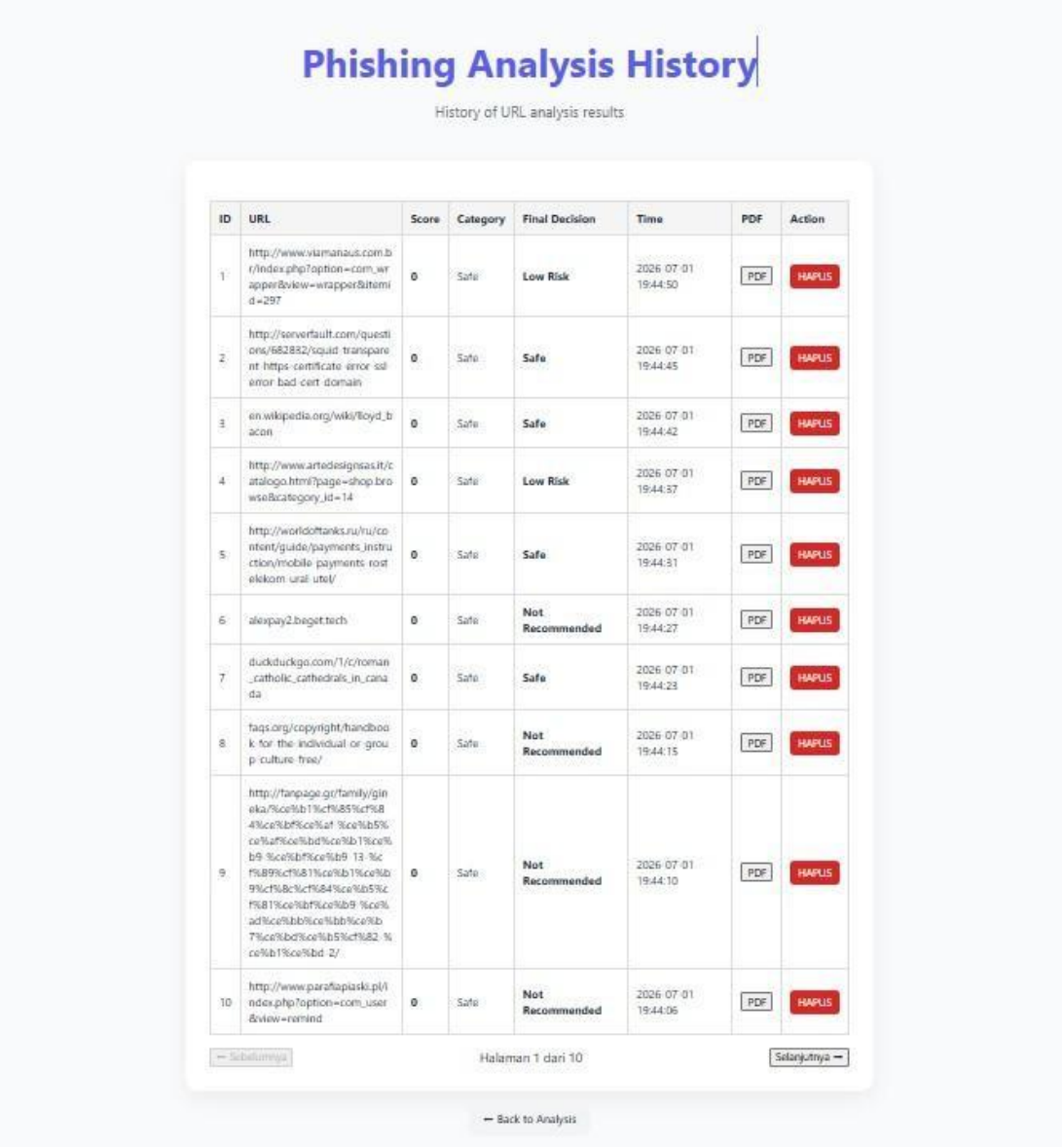


Figure 4.13 Analysis History

Figure 4.13 Analysis History shows the test results of the Analysis History feature in the Phishing Link Forensic Analyzer system. The test was conducted by opening the analysis history page after several URLs had been successfully processed by the system. This page displays all analysis results stored in the database, including the analyzed URLs, risk scores, risk categories, final decisions, analysis times, and access to the generated PDF reports. Additionally, the system provides a feature to delete history data via the Delete button. Based on the test results, all data previously stored in the database was successfully retrieved in full and matched the results of the analyses that had been performed. This indicates that the mechanism for retrieving data from the database and presenting information on the user

interface is functioning properly. Thus, the Analysis History feature helps users track, document, and manage the results of previous phishing analyses

4.3.8 Export PDF

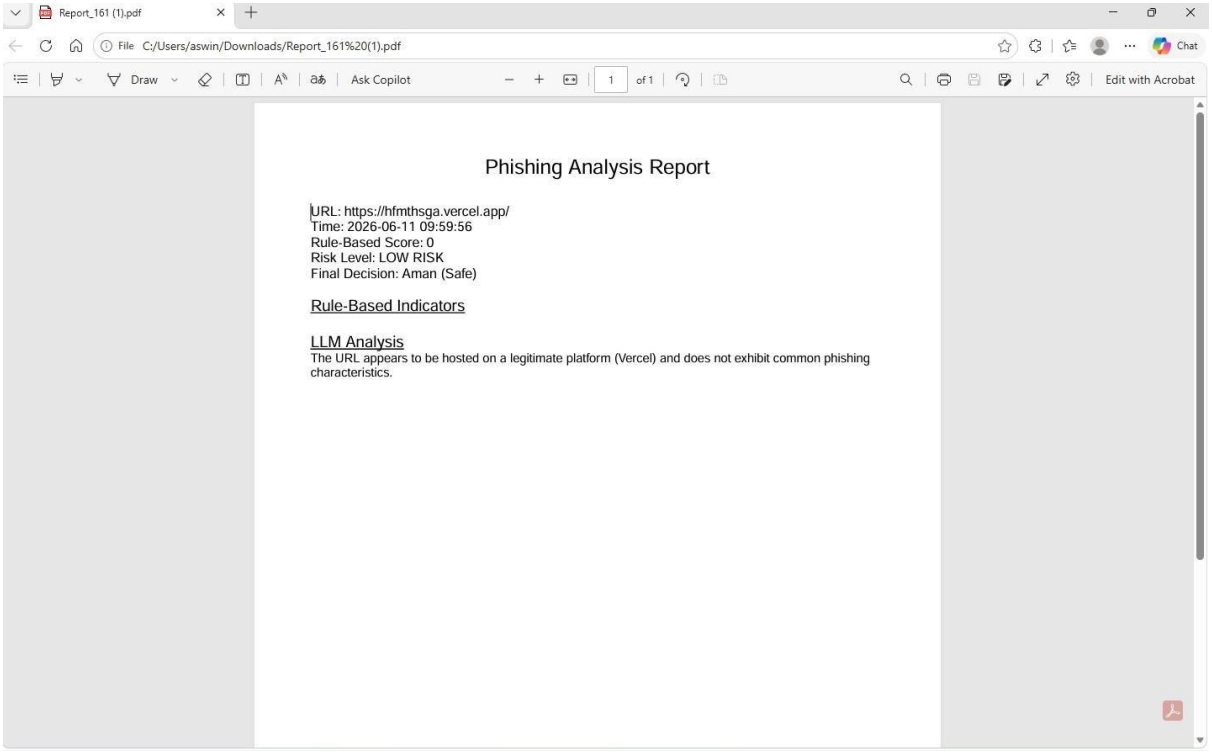


Figure 4.14 Export PDF

Figure 4.14 Export PDF shows the results of testing the Export PDF feature in the Phishing Link Forensic Analyzer system. The test was conducted by clicking the “Export Result to PDF” button after the URL analysis process was completed. The system then generated a PDF document containing the analysis results, including the analyzed URL, risk score, final decision, detected rule-based indicators, and an explanation of the analysis results from the Large Language Model (LLM). Based on the test results, the PDF file was successfully created and could be opened properly using a PDF reader application. The information displayed in the report matches the analysis results found in the system. This indicates that the report export feature is functioning as designed and is capable of providing documentation of the analysis results that can be saved or shared with users.

4.3.9 Docker Backend

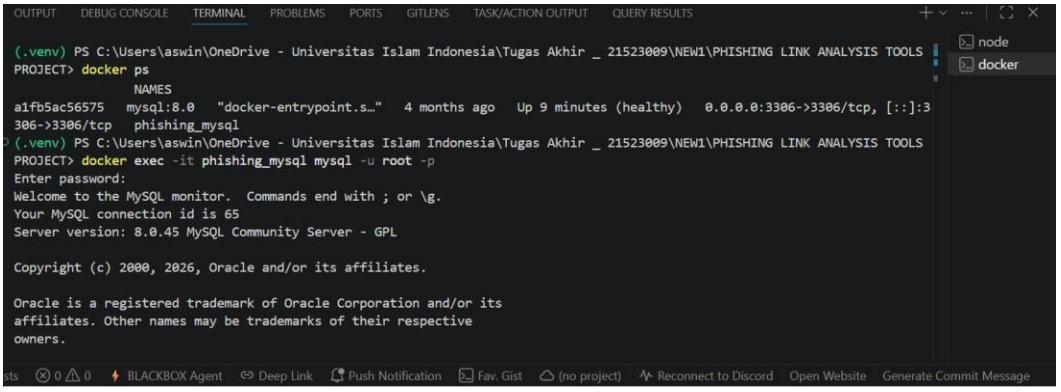


Figure 4.15 Docker Backend

Figure 4.15 Docker infrastructure testing was conducted to ensure that the MySQL database service used by the system runs properly in a Docker environment. The testing was performed using the `docker ps` command to check the container status and the `docker exec` command to access the MySQL service running inside the container. Based on the test results, the `phishing_mysql` container successfully ran with a “healthy” status and was accessible via the terminal. Additionally, the system successfully connected to the MySQL Community Server version 8.0.45 without encountering any connection issues. These results indicate that the Docker environment used to run the database is functioning properly and is capable of supporting the phishing detection system’s operations stably.

4.3.10 Results Display

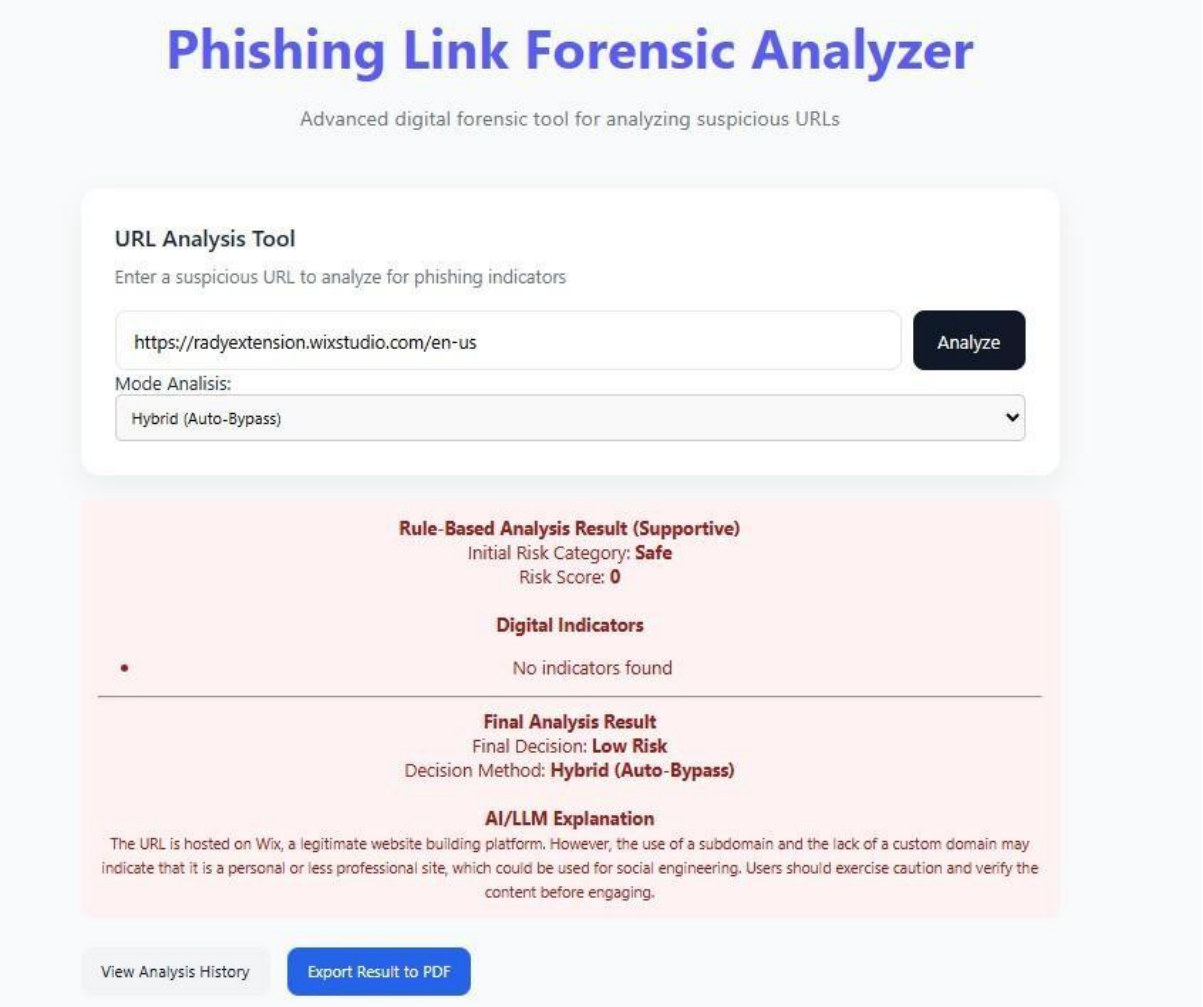


Figure 4.16 Results Display

Figure 4.16 illustrates the comprehensive diagnostic output provided by the Results Display feature when evaluating the URL (https://radyextension.wixstudio.com/en-us) under the "Hybrid (Auto-Bypass)" operational mode. Following the completion of the digital forensic scan, the system successfully presents a structured, multi-layered summary of the findings. The rule-based engine initially assesses the input as "Safe" with a risk score of 0, explicitly confirming the absence of explicit static indicators. Subsequently, the LLM module evaluates the contextual characteristics of the URL, identifying that the domain is hosted on a legitimate website-building platform (Wix) while highlighting potential risks associated with default subdomain structures and the lack of a custom domain, which may indicate a less professional site potentially susceptible to social engineering. Synthesizing these insights, the system generates a final decision of "Low Risk" via the Hybrid (Auto-Bypass) decision method. Furthermore, the interface provides integrated options at the bottom such as viewing the

analysis history or exporting the results to a PDF report demonstrating that the results display feature functions accurately, transparently, and comprehensively.

4.4 System Effectiveness

Based on the test results, the combination of rule-based methods and LLMs yielded better results than using a single method. The rule-based method is able to quickly detect known phishing indicators, while LLM functions to perform advanced analysis of URL patterns more thoroughly. This approach allows the system to not only rely on the number of phishing indicators found, but also to consider the relationships between the indicators as well as the overall URL structure patterns. Thus, the system is able to provide clearer and more in-depth outcome recommendations. The main advantage of this system is its ability to detect phishing patterns that are not defined in the system rules (unpredictable phishing). Under certain conditions, URLs that don't have many phishing indicators can still be recognized as phishing by LLMs based on suspicious URL patterns and structures. In addition, the integration between rule-based and LLM also helps reduce the possibility of false detection (false positive and false negative). The system is able to re-validate the results of the initial analysis so that the final decision made is more accurate and adaptive. However, some limitations are still found in the implementation of the system. In some conditions, the system still generates overly secure classifications of URLs that are actually relatively dangerous. In addition, the analysis process time in the LLM module is also relatively longer than rule-based analysis because it requires a more complicated analysis process.

4.4.1 Confusion Matrix Distribution of Test Data Hybrid (Auto-Bypass)

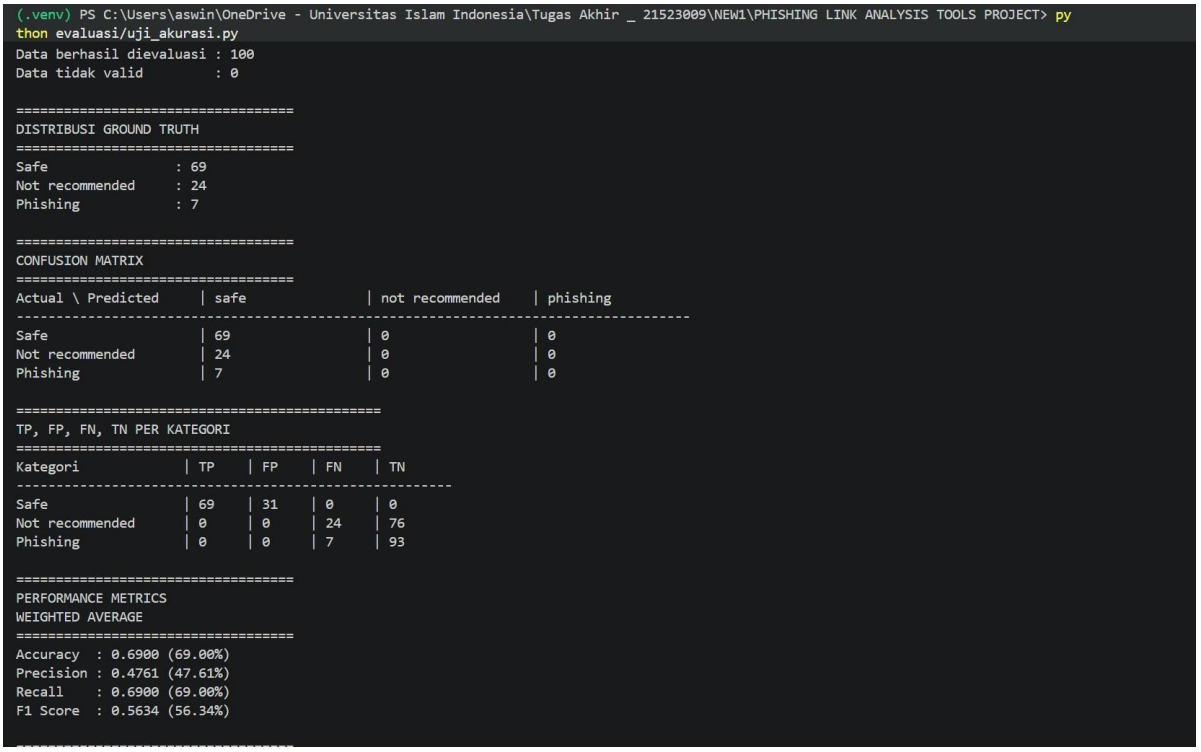


Figure 4.17 Confusion Matrix

The performance evaluation of the Hybrid (Auto-Bypass) configuration was conducted using 100 URL samples. Before evaluation, the original dataset labels were normalized into three evaluation categories, namely Safe, Not Recommended, and Phishing. This normalization was applied because the developed system produces four user-facing risk levels, while the evaluation process uses three standardized categories after label mapping. The resulting confusion matrix is presented in Table 4.1.

Table 4.1 Confusion Matrix Result

Actual Class \ Predicted	**Safe**	**Not Recommended**	**Phishing**
Safe	69	0	0
Not Recommended	24	0	0
Phishing	7	0	0

Based on the confusion matrix, all 100 tested URLs were predicted as Safe by the evaluated Hybrid (Auto-Bypass) configuration. The classification distribution can be explained as follows.

1. Safe URLs

A total of 69 URLs were categorized as Safe according to the ground-truth labels. All 69 Safe URLs were correctly classified as Safe by the system. Therefore, the system achieved 69 True Positives for the Safe category, with no Safe URLs classified as Not Recommended or Phishing. This indicates that the system was able to consistently identify the URLs categorized as Safe within the evaluated dataset.

2. Not Recommended URLs

A total of 24 URLs belonged to the Not Recommended ground-truth category. However, all 24 URLs were classified as Safe by the system. Therefore, the system obtained 0 True Positives for the Not Recommended category and 24 False Negatives. This result indicates that the system failed to identify the suspicious URLs belonging to this category during the evaluation.

3. Phishing URLs

A total of 7 URLs belonged to the Phishing ground-truth category. All 7 Phishing URLs were classified as Safe. Consequently, the system obtained 0 True Positives for the Phishing category and 7 False Negatives. This result is particularly important because it indicates that the evaluated configuration did not successfully identify any of the Phishing URLs contained in the test dataset.

4. Overall Classification Distribution

The confusion matrix shows a strong prediction bias toward the Safe category. Although all 69 Safe URLs were correctly identified, 31 URLs that should have been classified as potentially dangerous were also predicted as Safe, consisting of 24 Not Recommended URLs and 7 Phishing URLs.Therefore, the main classification problem observed in this experiment is the failure to distinguish malicious or suspicious URLs from legitimate URLs. This condition produces a relatively high number of False Negatives for threat detection.

The result demonstrates that the Hybrid (Auto-Bypass) configuration was effective in maintaining correct classification for the Safe category but was unable to adequately identify the Not Recommended and Phishing categories in this particular test configuration.

4.4.2 Evaluation Metric Value Analysis

Table 4.1 Evaluation Metric Value Analysis

Metric	Value
Accuracy	69.00%
Precision	47.61%
Recall	69.00%
F1-Score	56.34%

The evaluation results presented in Table 4.2 show that the Hybrid (Auto-Bypass) configuration achieved an overall Accuracy of 69.00%. This means that 69 out of 100 tested URLs were classified correctly according to the standardized evaluation labels. The Precision value of 47.61% indicates that the overall reliability of the generated class predictions remains limited when considering the distribution across all evaluation categories. The Recall value of 69.00% is strongly influenced by the successful classification of the Safe category, which contains the largest number of samples in the dataset.

The resulting F1-Score of 56.34% represents the harmonic mean between Precision and Recall. Although the Accuracy and Recall values reach 69.00%, the lower Precision and F1-Score indicate that the performance is not evenly distributed across the three evaluation categories. The classification report further illustrates this condition. The Safe category achieved a Precision of 0.69, Recall of 1.00, and F1-Score of 0.82 with 69 samples. In contrast, both the Not Recommended and Phishing categories obtained 0.00 Precision, 0.00 Recall, and 0.00 F1-Score, because none of the samples from these two categories were correctly predicted.

These results indicate that the overall Accuracy of 69.00% should not be interpreted as evidence that the system successfully detects phishing threats. Instead, the result is largely influenced by the correct classification of the majority Safe class. The absence of correctly detected Phishing samples demonstrates that the system still requires improvement in identifying malicious URLs.

4.4.3 Methode Performance Comparison

Table 4.2 Method Performance Comparison

Method	Accuracy	Precision	Recall	F1-Score
Rule-Based	69.00%	47.61%	69.00%	56.34%
Large Language Model (LLM)	71.00%	65.41%	71.00%	62.55%
Hybrid (Auto-Bypass)	69.00%	47.61%	69.00%	56.34%
Hybrid (Full Validation)	71.00%	61.59%	71.00%	61.14%

As shown in Table 4.3, the performance comparison evaluates the four detection approaches implemented in the proposed system. The Rule-Based method achieved an Accuracy of 69.00%, Precision of 47.61%, Recall of 69.00%, and F1-Score of 56.34%. These results indicate that the rule-based approach provides a deterministic baseline for identifying suspicious URL characteristics based on predefined structural indicators.

The Large Language Model (LLM) approach achieved the highest overall performance in terms of Accuracy, Precision, Recall, and F1-Score among the evaluated methods, with an Accuracy of 71.00%, Precision of 65.41%, Recall of 71.00%, and F1-Score of 62.55%. Compared with the Rule-Based method, the LLM achieved a 2.00 percentage-point improvement in Accuracy, 17.80 percentage points in Precision, 2.00 percentage points in Recall, and 6.21 percentage points in F1-Score. This indicates that contextual analysis performed by the LLM provided a more balanced classification performance in the evaluated dataset.

The Hybrid (Auto-Bypass) configuration obtained the same performance values as the Rule-Based method, namely 69.00% Accuracy, 47.61% Precision, 69.00% Recall, and 56.34% F1-Score. This indicates that, in the evaluated Auto-Bypass configuration, the integration of the LLM did not produce an additional improvement over the Rule-Based baseline. The result suggests that the automatic bypass mechanism may have limited the contribution of the LLM during part of the classification process.

Meanwhile, the Hybrid (Full Validation) configuration achieved an Accuracy of 71.00%, Precision of 61.59%, Recall of 71.00%, and F1-Score of 61.14%. Compared with the Hybrid (Auto-Bypass) configuration, the Full Validation approach improved Accuracy by 2.00 percentage points, Precision by 13.98 percentage points, Recall by 2.00 percentage points, and F1-Score by 4.80 percentage points.

Overall, the comparison shows that the LLM method achieved the highest F1-Score of 62.55%, while the Hybrid (Full Validation) configuration achieved the second-highest F1-Score of 61.14%. This result indicates that applying full validation provides a more balanced performance than the Auto-Bypass configuration. However, the difference between the LLM and Hybrid (Full Validation) approaches also shows that combining Rule-Based detection with LLM analysis does not automatically guarantee superior performance. The effectiveness of the hybrid architecture depends on how the rule-based results and LLM analysis are integrated within the decision-making process.

4.4.4 Label Normalization and Risk Mapping

The label normalization used in the evaluation was defined in Chapter III before the performance testing was conducted. The developed system generates four user-facing risk levels, namely Safe, Low Risk, Not Recommended, and Phishing. However, the original dataset contains four ground-truth categories, namely Benign, Defacement, Malware, and Phishing. Therefore, the dataset labels and system predictions were normalized into three standardized evaluation categories: Safe, Not Recommended, and Phishing.

For the ground-truth labels, Benign was mapped to Safe, while Defacement and Malware were grouped into Not Recommended. Phishing remained mapped to Phishing. For system predictions, both Safe and Low Risk were normalized to Safe, while Not Recommended and Phishing remained in their respective categories. These mappings were defined in Tables 3.9 and 3.10 and were applied before constructing the confusion matrix and calculating Accuracy, Precision, Recall, and F1-Score.

Table 4.3 Table Label Normalization and Risk Mapping

Category	**Original Label / System Output**	**Normalized Evaluation Label**
Ground Truth	Benign	Safe
Ground Truth	Defacement	Not Recommended
Ground Truth	Malware	Not Recommended
Ground Truth	Phishing	Phishing

System Prediction	Safe	Safe
System Prediction	Low Risk	Safe
System Prediction	Not Recommended	Not Recommended
System Prediction	Phishing	Phishing

The normalization process ensures consistency between the original dataset labels and the classification taxonomy implemented by the proposed system. The mapping was established before the evaluation process, rather than being introduced after the confusion matrix was generated. For the ground-truth data, Benign was normalized to Safe, while Defacement and Malware were normalized to Not Recommended because both categories represent non-phishing malicious or unsafe conditions in the evaluation taxonomy. Phishing remained classified as Phishing.

For system predictions, Safe and Low Risk were both normalized to Safe for evaluation purposes. This normalization is necessary because the system provides Low Risk as a user-facing risk level, whereas the evaluation uses three standardized categories. Consequently, the confusion matrix and evaluation metrics were calculated using the three normalized categories: Safe, Not Recommended, and Phishing.

4.4.5 Analysis of Classification Errors

The confusion matrix results indicate that the primary classification problem in the evaluated Hybrid (Auto-Bypass) configuration was the failure to identify the Not Recommended and Phishing categories. Based on the latest evaluation results, all 69 Safe samples were classified as Safe, while all 24 Not Recommended samples and all 7 Phishing samples were classified as Safe. Therefore, the system produced 69 correct predictions out of 100 samples, resulting in an Accuracy of 69.00%.

For the Safe category, the system correctly classified all 69 samples as Safe. This resulted in a True Positive value of 69 for the Safe category and a Recall of 100%. However, the high number of Safe predictions also indicates that the system tends to favor the Safe category when processing URLs.

For the Not Recommended category, none of the 24 samples were correctly classified as Not Recommended. All 24 samples were instead classified as Safe. Consequently, the Not Recommended category obtained a True Positive value of 0 and a False Negative value of 24. The same pattern occurred for the Phishing category, where all 7 Phishing URLs were classified as Safe. Therefore, the Phishing category obtained a True Positive value of 0 and a False Negative value of 7.

Importantly, the latest confusion matrix does not show False Positive predictions for the Not Recommended or Phishing categories, because no samples were classified as either of these two categories. The primary error pattern is therefore not an excessive warning rate, but rather a bias toward the Safe classification, which causes malicious or suspicious URLs to be overlooked.

The first factor contributing to this classification behavior is the limitation of URL-based analysis. The developed system primarily analyzes URL string structures and predefined rule indicators without directly analyzing web page content, SSL certificates, server logs, domain age, DNS records, or external domain reputation. As a result, sophisticated phishing URLs that use clean and legitimate-looking structures may not trigger enough static indicators to reach the phishing threshold.

The second factor is the structural similarity between legitimate URLs and modern phishing URLs. Phishing websites can use legitimate-looking domain structures, regional top-level domains, HTTPS connections, and ordinary URL paths to reduce visible suspicious characteristics. For example, a phishing domain may imitate a legitimate brand while maintaining a structurally simple URL. In such cases, URL-based rules may fail to generate sufficient evidence for a higher risk classification.

The third factor relates to the contextual interpretation performed by GPT-4o-mini. The LLM acts as a contextual decision component by considering the URL and the indicators generated by the Rule-Based module. Although this approach is intended to improve the adaptability of the detection system, the LLM may produce conservative decisions when the available evidence is limited. When a URL does not contain obvious indicators such as an IP address, URL shortener, excessive subdomains, or suspicious symbols, the model may interpret the URL as relatively legitimate.

Furthermore, the current implementation does not provide the LLM with external contextual evidence such as domain age, DNS information, website content, certificate information, or domain reputation. Consequently, the LLM must make its decision primarily from the URL structure and the available Rule-Based indicators. This limitation can contribute to False Negative classifications, particularly for phishing URLs designed to closely resemble legitimate websites.

Overall, the observed classification errors demonstrate that the current system has a strong tendency to classify URLs as Safe when insufficient evidence is available to support a higher-risk classification. Although this behavior results in a relatively high Recall for the Safe

category, it significantly reduces the detection capability for the Not Recommended and Phishing categories. Future development should therefore focus on improving heuristic weighting, refining the LLM prompt and decision mechanism, and integrating additional contextual evidence such as domain reputation, DNS records, SSL certificate information, website content analysis, and domain age to improve the detection of malicious URLs.

4.5 Conclusion

Based on the implementation and evaluation results, it can be concluded that the proposed hybrid phishing detection system successfully integrates deterministic Rule-Based Detection with the GPT-4o-mini Large Language Model to perform URL-based analysis. The Rule-Based module identifies structural URL anomalies using predefined heuristic rules, while the LLM provides contextual interpretation of URL characteristics that cannot be fully captured by static rules alone.

The evaluation process was conducted using a dataset of 100 URL samples with known ground-truth labels. To maintain consistency between the dataset and the proposed system, the original dataset labels were normalized before evaluation. Benign was mapped to Safe, Defacement and Malware were mapped to Not Recommended, and Phishing remained mapped to Phishing. System predictions of Safe and Low Risk were normalized to Safe, while Not Recommended and Phishing retained their respective categories. Consequently, the confusion matrix and evaluation metrics were calculated using three standardized evaluation categories: Safe, Not Recommended, and Phishing.

Based on the method comparison results, the Hybrid (Full Validation) configuration achieved an Accuracy of 71.00%, Precision of 61.59%, Recall of 71.00%, and F1-Score of 61.14%. These results indicate that the Full Validation configuration provided the strongest overall performance among the evaluated hybrid configurations, while the Hybrid (Auto-Bypass) configuration achieved an Accuracy of 69.00%, Precision of 47.61%, Recall of 69.00%, and F1-Score of 56.34%.

The latest confusion matrix for the Hybrid (Auto-Bypass) configuration also revealed an important classification behavior. All 69 Safe samples were correctly classified as Safe, while all 24 Not Recommended samples and all 7 Phishing samples were incorrectly classified as Safe. This demonstrates that the system has a strong tendency toward the Safe classification and still faces difficulties in detecting malicious URLs that do not contain sufficiently distinctive structural indicators.

The evaluation further demonstrates that URL-based analysis has inherent limitations when dealing with sophisticated phishing URLs. URLs that closely resemble legitimate websites may evade both static rule detection and contextual LLM analysis when additional evidence, such as domain reputation, DNS information, SSL certificates, website content, or domain age, is unavailable. Therefore, the obtained results indicate that the proposed system provides a functional and explainable foundation for phishing URL analysis, but further improvements are required to increase its capability to identify sophisticated phishing and other malicious URLs.

Furthermore, the developed web-based application successfully provides automated URL risk classification, Rule-Based indicator analysis, LLM-based contextual explanation, database-backed analysis history, and dynamic PDF report generation. The integration of these components demonstrates that the proposed system can function as a supporting tool for phishing awareness and preliminary digital forensic analysis. Future development should focus on incorporating additional external evidence, improving rule weighting and LLM prompt calibration, and expanding the evaluation dataset to improve the robustness and generalization of the proposed system.

CHAPTER V
CONCULSION

5.1 Conslusion

Based on the results of research, implementation, and testing that has been conducted, it can be concluded that a phishing detection system based on a combination of rule-based methods and Large Language Model (LLM) has been successfully developed and is able to carry out the URL-based phishing identification process properly and systematically. This research is motivated by the increasing threat of phishing that uses fake websites to obtain sensitive user information, where conventional rule-based detection methods still have limitations in recognizing new phishing patterns that are not defined in the system rules.

The rule-based method in this study was used to detect predictable phishing indicators, such as the use of IP addresses as domains, the presence of the "@" symbol, abnormal URL lengths, the use of sensitive keywords, the use of URL shorteners, the number of subdomains, and the use of hyphens on domains. The results of the rule-based analysis are then forwarded to the GPT-4o-mini-based Large Language Model (LLM) module to conduct further analysis of URL patterns more thoroughly and interpretively.

The implementation of LLM allows the system to understand the relationships between indicators as well as detect unpredictable phishing patterns that cannot be recognized using rule-based methods alone. With this hybrid approach, the system not only generates a security classification of URLs, but is also able to provide a more in-depth explanation of the results of the analysis based on the context and pattern of the analyzed URLs.

The test results showed that the integration of rule-based methods and LLMs was able to increase the effectiveness of the system in detecting phishing compared to using a single method. The system also successfully executes other supporting features, such as storing the analyzed data to the database, generating the analysis report in PDF format, and implementing the backend and database using Docker to maintain the consistency of the system environment and improve the stability of application services.

Thus, the developed system can be one of the supporting solutions in the URL-based phishing identification process in the field of cybersecurity, especially in helping to detect

predictable and unpredictable phishing patterns through a combination of rule-based analysis and Large Language Model (LLM)-based analysis.

5.2 Suggestions

Although the system developed has been able to perform the phishing detection function well, there are still several things that can be developed in future research.

1. Further research is suggested to develop a more diverse method of identifying digital artifacts, not only limited to URL analysis, but also including server logs, web page content, metadata, and network traffic to make the process of identifying phishing activity more complete and accurate.
2. The next development of the system can be focused on improving the ability to analyze phishing patterns by adding more adaptive analysis methods, such as the use of the latest phishing datasets or fine-tuning LLM models, so that the system is able to recognize new phishing patterns that continue to develop.
3. Further research is recommended to integrate phishing detection systems with other cybersecurity technologies, such as browser extensions or network monitoring systems, so that the process of early detection and prevention of phishing can be carried out automatically and in real-time.

REFERENCE

Adiputra, R., & Setiawan, A. (2023). Web-Based Malicious URL Classification Using Improved Random Forest. *Journal of Computer Science and Information Technology*, 11(2), 85–94.

Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J. P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. *Electronics (Switzerland)*, 9(9), 1–24. https://doi.org/10.3390/electronics9091514

Aryanti, R., & Nabila, Z. (2023). Phishing URL Detection Using NLP with SVM and TF-IDF Techniques. *International Journal of Cyber Security*, 5(1), 45–52.

Elder, J. (2009). *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007). A framework for detection and measurement of phishing attacks. *Proceedings of the 2007 ACM Workshop on Rapid Malicious Code Detection*, 1–8. https://doi.org/10.1114/1234567

Harahap, A. D., Juardi, D., & Irawan, A. S. Y. (2024). RANCANG BANGUN SISTEM PENDETEKSI LINK PHISHING MENGGUNAKAN ALGORITMA RANDOM FOREST BERBASIS WEB. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3). https://doi.org/10.23960/jitet.v12i3.4858

Khonji, M., & Iraqi, Y. (2011). Lexical URL analysis for discriminating phishing web sites. *2011 International Conference on Innovations in Information Technology*, 179–184. https://doi.org/10.1109/INNOVATIONS.2011.5893813

Lukito, S., & Handaya, B. T. (2022). Phishing URL Detection Using XGBoost and Hyperparameter Tuning. *Journal of Software Engineering and Applications*, 14(4), 112–121.

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2011). Lexical feature based phishing URL detection using statistical methods. *Communications of the ACM*, 54(9), 92–100. https://doi.org/10.1145/1995376.1995402

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, 25(2), 443–458. http://eprints.hud.ac.uk/id/eprint/19220/

Moore, T., & Clayton, R. (2011). The Impact of Public Information on Phishing Attack and Defense. *Conference on Economics of Information Security (CEIS)*, 3(2), 204–215.

Noori, N., Bawanthad, V., Pakhare, M., Agrawal, R., & Kimbahune, V. (2023). Phishing URL Detection using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, 11(5), 3645–3648. https://doi.org/10.22214/ijraset.2023.52342

Pradana, A., et al. (2019). Rule Based Attribute Checking for Static Phishing Identification. *Journal of Cyber Forensics*, 7(3), 130–139.

Salvi, S., et al. (2021). Detection of Phishing Websites Using Machine Learning Algorithms. *International Journal of Computer Applications*, 183(14), 25–31. https://doi.org/10.5120/ijca2021921345

APPENDIX