Jurnal Sistem Informasi dan Teknik Informatika
Vol. 1, No. 2, August 2023, pp. 54~61
ISSN: 3031-2698 (Online)

Naive Bayes-based Visualization of Disease-related Data in Muara Telang Public Health Centers

Nopri Yansah
University of Indo Global Mandiri, Palembang, Indonesia

Article Info

Article history:

Received Apr 25th, 2023
Revised May 21th, 2023
Accepted Jun 15th, 2023

Corresponding Author:

Nopri Yansah
University of Indo Global Mandiri, Indonesia
Email: nopri_yigm@gmail.com

Abstract

This study addresses the imperative need for active and efficient management of Community Health Centers (Puskesmas) in Indonesia, as mandated by the Minister of Health's Regulation. The role of these centers in managing and planning health development within their work areas is crucial for comprehensive, integrated, acceptable, and affordable health efforts. The utilization of visualization, particularly through computer technology, emerges as a vital tool for conveying complex health data to the public, facilitating quicker understanding and informed decision-making. Focusing on the Sumber Marga Telang Health Center in South Sumatra Province, the research employs a structured waterfall model for system development. The Naive Bayes algorithm is utilized to classify internal and eye diseases, offering a practical solution to the challenge of conveying disease data to the public effectively. In conclusion, this research provides a comprehensive approach to enhancing health information dissemination and data visualization at community health centers, contributing significantly to public awareness and preventive healthcare efforts.

Keywords: *Community Health Centers, Health Management, Data Visualization, Naive Bayes Algorithm, Public Health Awareness*

Abstrak

Kajian ini menjawab kebutuhan mendesak akan pengelolaan Pusat Kesehatan Masyarakat (Puskesmas) yang aktif dan efisien di Indonesia, sebagaimana diamanatkan oleh Peraturan Menteri Kesehatan. Peran pusat-pusat tersebut dalam mengelola dan merencanakan pembangunan kesehatan di wilayah kerjanya sangat penting untuk mencapai upaya kesehatan yang komprehensif, terpadu, dapat diterima, dan terjangkau. Pemanfaatan visualisasi, khususnya melalui teknologi komputer, muncul sebagai alat penting untuk menyampaikan data kesehatan yang kompleks kepada masyarakat, memfasilitasi pemahaman yang lebih cepat dan pengambilan keputusan yang tepat. Berfokus pada Puskesmas Sumber Marga Telang Provinsi Sumatera Selatan, penelitian ini menggunakan model air terjun terstruktur untuk pengembangan sistem. Algoritme Naive Bayes digunakan untuk mengklasifikasikan penyakit dalam dan mata, menawarkan solusi praktis terhadap tantangan penyampaian data penyakit kepada publik secara efektif. Kesimpulannya, penelitian ini memberikan pendekatan komprehensif untuk meningkatkan penyebaran informasi kesehatan dan visualisasi data di pusat kesehatan masyarakat, sehingga memberikan kontribusi yang signifikan terhadap kesadaran masyarakat dan upaya perawatan kesehatan preventif.

Kata kunci: Puskesmas, Manajemen Kesehatan, Visualisasi Data, Algoritma Naive Bayes, Kesadaran Kesehatan Masyarakat

1. INTRODUCTION

Based on the Regulation of the Minister of Health of the Republic of Indonesia Number 44 of 2016 concerning Community Health Center Management guidelines, to implement health efforts, both public health and individual level efforts, active and efficient management of Community Health Centers (Puskesmas) is required. The function of the Community Health Center is to manage and plan health development in its work area. The Community Health Center is a functional organization that organizes health efforts that are comprehensive, integrated, acceptable, and affordable to the community, with costs borne by the government and the community [1], [2]. Visualization is a human effort to describe certain intentions in the form of information that is easier to understand. Usually, nowadays, humans use to create visualizations using computer technology. Information in visual form can be understood quickly, allowing users to see data connectivity, and making it easier to make decisions.

Sumber Marga Telang Health Center is a health center located in Muara Telang, Sumber Marga Telang District Banyuasin Regency in South Sumatra Province. Muara Telang Health Center is a sub-district health center consisting of nine villages with an area of 956.6 km with a population of 25,480 people. The Sumber Marga Telang District Health Service is one of the government health service agencies that must monitor public health and the environment. The monitoring results are reported in the form of disease data, including internal and eye diseases. The data reported is in the form of raw data, namely in the form of tables and text. This condition causes difficulties in conveying information on data from monitoring a disease case to the public. This causes people to have limited knowledge about the diseases with the highest numbers every year. Based on the problems above, the author is interested in addressing these problems by providing a solution in the form of data visualization system and health information, including disease data. Accurate visualization in the form of graphs and tables will facilitate health prevention and promotion work in the community.

2. METHOD

Figure 1 illustrates the research methodology employed by the author to fulfill the initial objectives. The system development phase utilized in this study is the waterfall model, known for its characteristic requiring completion of each phase before advancing to the subsequent one. This approach follows a structured and linear flow within software development [3], [4], [5]. The process comprises a sequence of phases that must be finished in order, where the conclusion of each phase is reliant on the prior one. Below are several primary stages within the research model:

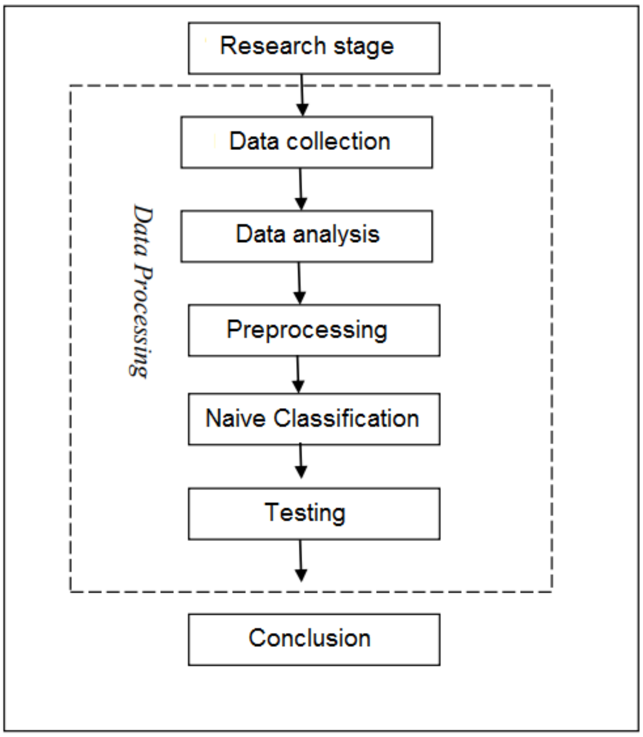


Figure 1 – System development model

1. Data Collection: This initial stage involves gathering relevant data that will be used for analysis. This could involve sourcing information from various databases, conducting surveys, scraping websites, or utilizing existing datasets.

2. Data Analysis: Once the data is collected, the next step is to analyze it. This involves exploring the data to understand its structure, patterns, and relationships. Techniques like statistical analysis or visualization tools are commonly used in this stage to gain insights into the data.
3. Preprocessing: Data collected may have inconsistencies, missing values, outliers, or other issues. Preprocessing involves cleaning and preparing the data for further analysis. This step might include handling missing values, normalizing data, or transforming it to a more suitable format for analysis.
4. Naive Classification: Naive classification refers to the application of a simple algorithm or model to the data for classification purposes. The "naive" aspect implies a straightforward approach, often used as a baseline for comparison with more sophisticated models. For instance, in machine learning, a naive classification method could be a basic rule-based or simple statistical model.
5. Testing: After applying the naive classification or any other method, it's essential to test its performance. This involves evaluating how well the model/classification method performs on new, unseen data. Metrics like accuracy, precision, recall, or F1 score might be used to assess the model's effectiveness.
6. Conclusion: Based on the results obtained from testing, researchers draw conclusions. This involves interpreting the findings, discussing their implications, and determining whether the initial hypothesis or objective has been supported or refuted. It often involves discussing the limitations of the study and potential future research directions.

This sequence of steps provides a structured approach to conducting research or analysis, particularly in the realm of data-driven studies. It helps ensure that data is properly handled, analyzed, and interpreted to draw meaningful conclusions.

Naive Bayes is a simple probabilistic classification algorithm based on Bayes' theorem. It's called "naive" because of its assumption of feature independence, which means it assumes that the presence of one feature is independent of the presence of any other feature. Despite this oversimplified assumption, Naive Bayes often performs well in various real-world applications, particularly in text classification and spam filtering.

The fundamental equation behind Naive Bayes is Bayes' theorem is shown in Equation (1):

$$P(A \mid B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (1)$$

In the context of classification problems, here's how Naive Bayes works:

a. Prior Probability (P(A)): $P(A)$ represents the prior probability of class A occurring without considering any features. In a classification problem, it's the probability of a particular class occurring in the dataset without any knowledge of the features.
b. Likelihood (P(B|A)): $P(B|A)$ represents the likelihood of observing the features B given that the class A is true. This is calculated by assuming that the features are conditionally independent given the class.
c. Evidence (P(B)): $P(B)$ is the probability of observing the given set of features B. It acts as a normalization factor.

The steps involved in the Naive Bayes algorithm include:

1. Calculate Class Probabilities: Compute the prior probability $P(A)$ for each class in the dataset.
2. Calculate Conditional Probabilities: For each feature given each class, calculate the likelihood $P(B|A)$ using the training data. This involves assuming independence between features given the class.
3. Make Predictions: Given a new set of features, use Bayes' theorem to calculate the probability of each class given these features. The class with the highest probability is assigned as the predicted class.
The formula for predicting the class of a new instance using Naive Bayes is shown in Equation (2):

$$\text{argmax}_{c \in \text{classes}}(P(c) \times \prod_{i=1}^{n} P(xi \mid c)) \quad (2)$$

Where:

- $P(c)$ is the prior probability of class c.
- $P(xi|c)$ is the conditional probability of feature xi given class c.
- n is the number of features.

By applying this formula, Naive Bayes computes the probability of each class and selects the class with the highest probability as the prediction.

3. RESULTS AND DISCUSSION

The data obtained for this research uses quantitative data, namely historical data on internal medicine and eye disease, while the type of data is secondary data. Data collection aims to obtain the data needed for classification and testing, namely in the form of data on internal medicine and eye diseases. Data for internal diseases are shown in Table 1 and data for eye diseases are shown in Table 2.

Table 1 – Data of internal diseases

No	Code	Type of diseases	Gender		Total
			M	FM	
1	0102	Cholera	12	14	26
2	0161	Thypoid	8	16	24
3	0162	Dyspepsia	96	108	204
4	0163	Gastritis	209	128	337
5	0201	Tuberculosis	0	4	4
6	12	Hypertension	29	55	84
7	1301	Tonsillitis	2	10	12
8	1302	Respiratory infection	71	24	95
9	1361	Flu	30	31	61
10	1401	Pneumonia	10	9	19
11	1402	Bronchitis	41	18	59
12	1403	Asthma	5	4	9
13	2463	Anemia	21	26	47
14	2561	Rheumatism	10	24	34

Table 2 – Data of eye diseases

No	Code	Type of diseases	Gender		Total
			M	FM	
1	1002	Cataract	6	3	9
2	1003	Refraction	12	19	31

The author does not discuss all stages of the Naive Bayes method but goes directly to the stages that the author considers important to convey. After calculating the probability table, next, calculate the probability value for each class. The class probability is shown in Table 3:

Table 3 – Class probability

Class	
Internal diseases	Eye diseases
Internal diseases = 1115	Eye diseases = 40
P (Internal diseases) = 1115/1155 = 0.96	P (Eye diseases = 40/1155 = 0,03

Table 3 above is the probability value for each class based on existing data. In creating a Naive Bayes model, we first look for the probability of existing hypotheses, namely data on internal medicine and eye disease. The total data is 1155, for internal medicine it is 1115 and for eye disease, it is 40. After the probability for each hypothesis is known, the next step is to calculate the probability of the condition (probability X) based on the probability of each hypothesis (Probability H) called prior probability. The results of prior probability calculations using Naive Bayes are shown in Table 4.

Table 4 - Prior probability

Attribute	Number of cases	Internal disease	Eye disease	P(X|H)	
Total	1155	1115	40	Internal disease	Eye disease
Intestinal infections	591	591	0	0.530	0
Tuberculosis	4	4	0	0	0
Hypertension	184	184	0	0.165	0

Top respiratory tract infections	168	168	0	0.150	0
Bottom respiratory tract infections	87	87	0	0.078	0
Endocrine & Metabolic	81	81	0	0.072	0
Eyes and Adnexa	40	0	40	0	1

After finding the prior probability, next, determine the final probability value. How to find the final value can be seen below. For example, there is a patient with a type of intestinal infection, then:

P(X| Disease) = P (Intestinal infections = Y| Internal disease)
= 591/1115
= 0.53

P (X| Intestinal infections) = P (Eyes diseases = Y| Eyes disease)
= 0/40
= 0

Next, these scores are entered to get the final score:

P (X|hasil = Intestinal infections) P (Total number of disease)
= 591/1155
=0.5116

So the largest final probability scire is in the class of intestinal infectious diseases which are included in internal diseases.

= 591 x 1155/1
=682.60

The data classification stage aims to measure how well the classification process has been created. Classification using Naive Bayes is divided into 2 processes, namely the training and testing processes. The training process is used to produce sentiment analysis results which will later be used as a reference for classifying with testing or new raw data. This test uses 1155 disease data samples. For internal diseases 1115 data and for eye diseases 40 data then the researchers divided it into 2 parts, namely training data and testing data. After understanding how the Naive Bayes classifies disease-related data, the author started designing the proposed app by creating a UML diagram [6], [7]. The proposed system design includes Use case diagrams, Activity diagrams, Sequence diagrams, and Class diagrams.

3.1. Use Case diagram

Several things need to be described, namely actors and use cases. Actors are users who are connected to the system and can be people (indicated by their role and not their name/personnel). The actor is symbolized by the figure of a stick man with a noun at the bottom that states the role/system. Use cases are depicted with an ellipse symbol with the name of the active verb inside which states the activity from the actor's perspective [8], [9].

3.2. Activity diagram

An activity diagram is a description of function paths in an information system [10]. In full, the activity diagram defines where the system process starts, where it stops, what activities occur during the system process, and what sequence these activities occur in.

3.3. Class diagram

Class diagrams describe the types of objects in the system and the various static relationships that exist between them [11]. Class diagrams show the properties and operations of a class and the boundaries contained in the object relationships. Figure 2 shows the class diagram of the proposed system.

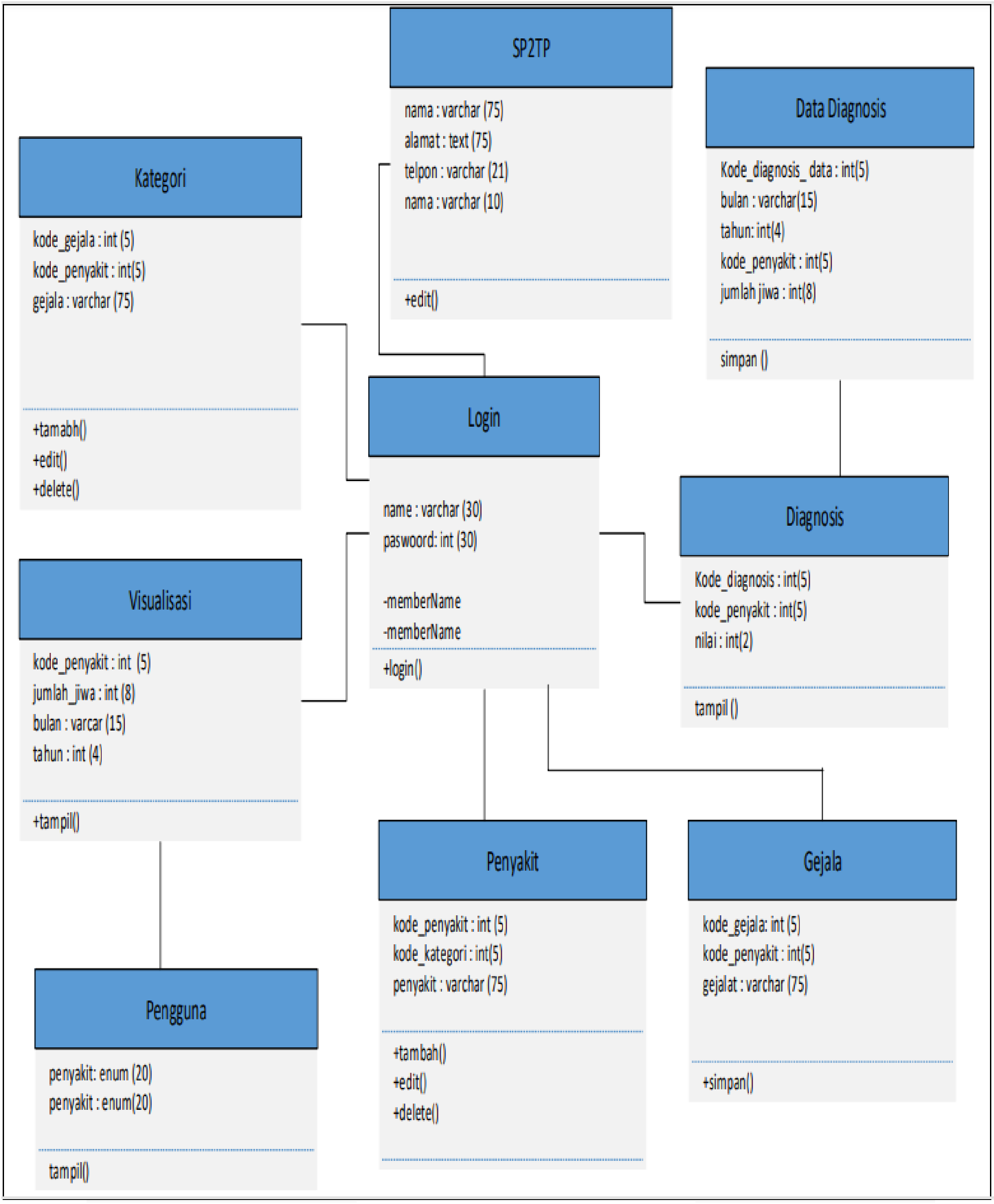
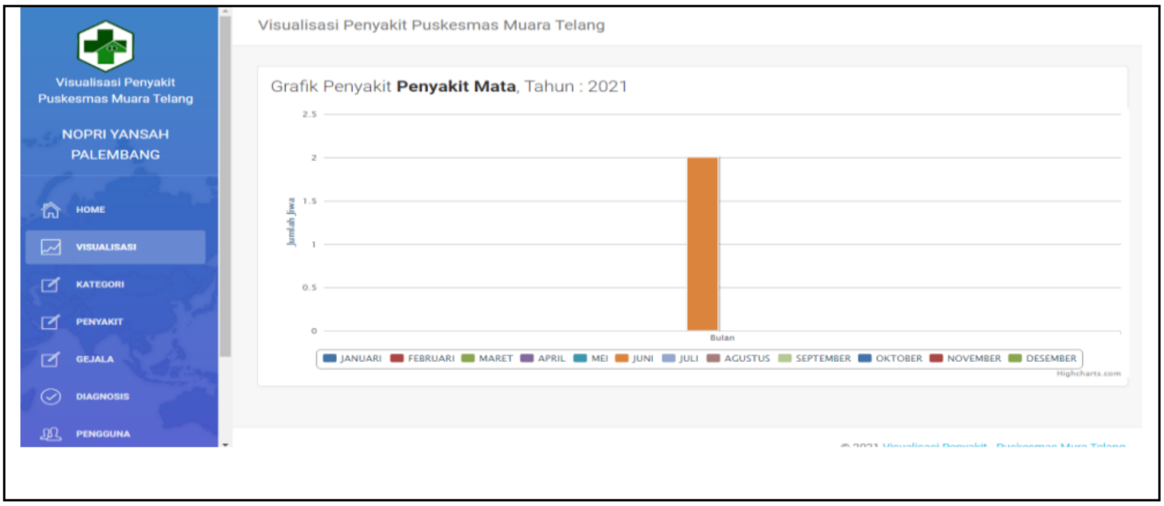


Figure 2 – Class Diagram

3.4. System Interface

A system interface refers to the point of interaction or communication between different systems, components, or software modules within a larger system or between separate systems. It defines how different parts of a system communicate, exchange data, or interact with each other. One example of the interface of the proposed system is the criteria comparison analysis page (Figure 3).



Visualisasi Penyakit Puskesmas Muara Telang

Grafik Penyakit **Penyakit Mata**, Tahun : 2021

Figure 3 – The Interface of the system

In the end, the author carries out Black Box testing of the app that has been built. Black Box testing focuses on the functional requirements of the software [12]-[15]. Thus, black box testing allows software engineers to obtain a set of input conditions that fully utilize all functional requirements for an app. Black box testing seeks to find errors in the following criteria: Incorrect or missing functions, Interface errors, Errors in data structure or database access, and Performance errors. Based on the test results, overall the app built meets all testing criteria, in line with expectations at the start of the study.

4. CONCLUSION

This study focused on addressing the challenges faced by Sumber Marga Telang Health Center in conveying disease data to the public through the implementation of a data visualization system and health information. The study followed a structured methodology, employing the waterfall model for system development. The Naive Bayes algorithm was applied to classify internal and eye diseases based on historical data. The results demonstrated successful classification and were utilized in the design of a proposed system, encompassing use case diagrams, activity diagrams, class diagrams, and a user interface. The black box testing of the developed application confirmed its functionality, meeting the predefined criteria. Overall, the research presents a comprehensive approach to improving health information dissemination and data visualization at the community health center, contributing to enhanced public awareness and preventive healthcare efforts.

ACKNOWLEDGEMENTS

The author would like to thank you Universitas Indo Global Mandiri.

REFERENCES

[1] S. Choi and T. Powers, “Engaging and informing patients: Health information technology use in community health centers,” *Int. J. Med. Inform.*, vol. 177, no. February, p. 105158, 2023, doi: 10.1016/j.ijmedinf.2023.105158.

[2] M. Fathoni *et al.*, “The preparedness of disaster among nurses in community health centers in rural areas during the COVID-19 pandemic in Malang City,” *Enferm. Clin.*, vol. 32, pp. S54–S57, 2022, doi: 10.1016/j.enfcli.2022.03.018.

[3] K. D. Prasetya, Suharjito, and D. Pratama, “Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 103–111, 2021, doi: 10.1016/j.procs.2020.12.014.

[4] T. Thesing, C. Feldmann, and M. Burchardt, “Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project,” *Procedia Comput. Sci.*, vol. 181, pp. 746–756, 2021, doi: 10.1016/j.procs.2021.01.227.

[5] A. A. S. Gunawan, B. Clemons, I. F. Halim, K. Anderson, and M. P. Adianti, “Development of e-butler: Introduction of robot system in hospitality with mobile application,” *Procedia Comput. Sci.*, vol. 216, no. 2019, pp. 67–76, 2022, doi: 10.1016/j.procs.2022.12.112.

[6] G. Bergström *et al.*, “Evaluating the layout quality of UML class diagrams using machine learning,” *J. Syst. Softw.*, vol. 192, p. 111413, 2022, doi: 10.1016/j.jss.2022.111413.

[7] H. Wu, “QMaxUSE: A new tool for verifying UML class diagrams and OCL invariants,” *Sci. Comput. Program.*, vol. 228, p. 102955, 2023, doi: 10.1016/j.scico.2023.102955.

[8] P. Danenas, T. Skersys, and R. Butleris, “Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams,” *Data Knowl. Eng.*, vol. 128, no. February, p. 101822, 2020, doi: 10.1016/j.datak.2020.101822.

[9] Meiliana, I. Septian, R. S. Alianto, Daniel, and F. L. Gaol, “Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm,” *Procedia Comput. Sci.*, vol. 116, pp. 629–637, 2017, doi:

10.1016/j.procs.2017.10.029.

[10] Z. Daw and R. Cleaveland, “Comparing model checkers for timed UML activity diagrams,” *Sci. Comput. Program.*, vol. 111, no. P2, pp. 277–299, 2015, doi: 10.1016/j.scico.2015.05.008.

[11] F. Chen, L. Zhang, X. Lian, and N. Niu, “Automatically recognizing the semantic elements from UML class diagram images,” *J. Syst. Softw.*, vol. 193, p. 111431, 2022, doi: 10.1016/j.jss.2022.111431.

[12] D. Felicio, J. Simao, and N. Datia, “Rapitest: Continuous black-box testing of restful web apis,” *Procedia Comput. Sci.*, vol. 219, no. 2022, pp. 537–545, 2023, doi: 10.1016/j.procs.2023.01.322.

[13] H. Bostani and V. Moonsamy, “EvadeDroid: A Practical Evasion Attack on Machine Learning for Black-box Android Malware Detection,” *Comput. Secur.*, p. 103676, 2021, doi: 10.1016/j.cose.2023.103676.

[14] F. Pagano, A. Romdhana, D. Caputo, L. Verderame, and A. Merlo, “SEBASTiAn: A static and extensible black-box application security testing tool for iOS and Android applications,” *SoftwareX*, vol. 23, p. 101448, 2023, doi: 10.1016/j.softx.2023.101448.

[15] C. Cronley *et al.*, “Designing and evaluating a smartphone app to increase underserved communities’ data representation in transportation policy and planning,” *Transp. Res. Interdiscip. Perspect.*, vol. 18, no. January, p. 100763, 2023, doi: 10.1016/j.trip.2023.100763.