Software Development for Swimmer Performance Prediction System Based on Physical Characteristics using XGBoost

Surya Pratama Tanzil [1]*, Tinaliah [2]*, Anugerah Widi [3]**
* Informatics Program Study, Universitas Multi Data Palembang
** Information System Program Study, Universitas Multi Data Palembang
suryapratamatanzil@mhs.mdp.ac.id [1], tinaliah@mdp.ac.id [2], widi@mdp.ac.id [3]

Article Info

Article history:

Received 2026-01-05
Revised 2026-01-23
Accepted 2026-01-29

Keyword:

Prediction Performance,
Swimmer,
Physical Characteristics,
Website,
XGBoost.

ABSTRACT

Swimmer performance assessment in Indonesia still largely depends on coaches' intuition, which may lead to subjective decisions and inconsistencies in training program planning, particularly in environments where frequent changes in coaches and sports administrators occur. The lack of structured and data-driven performance assessment tools further limits the continuity and objectivity of athlete development. This study aims to develop a web-based system capable of predicting swimmers' performance potential by estimating race times based on physical characteristics using the XGBoost model. The proposed system is designed to support coaches in identifying athlete performance potential in a more objective and data-driven manner. Model evaluation results indicate that the XGBoost model achieved an R^2 value of 0.9190, demonstrating a very high level of prediction accuracy, with an average prediction time of 7.036 seconds. Software testing results confirm that the system operates as intended and is able to present prediction outputs in the form of estimated swimming time, performance percentage, and performance classification into four categories: Very High, High, Medium, and Low. Furthermore, usability evaluation using the USE method yielded excellent results, with an average score of 88.16%.

I. INTRODUCTION

Swimming is a water-based sport activity that provides various benefits, including educational, recreational, health, therapeutic, rehabilitation, performance, and professional development purposes [1]. In addition to being a fitness activity, swimming is also a competitive sport that is frequently contested in various official events, ranging from regional competitions such as Pekan Olahraga Daerah (PORDA), national competitions such as Pekan Olahraga Nasional (PON), to international events such as the SEA Games and other championships [2]. The existence of multiple levels of competition in swimming indicates numerous opportunities for achieving sporting excellence. Athletes who demonstrate high performance in competitions have the opportunity to receive promotion and further development from official institutions such as Komite Olahraga Nasional Indonesia (KONI) [3].

However, the achievements of Indonesian swimming athletes to date have not yet met expectations in achieving sports achievements at the international level [4]. Based on medal statistics published on the official website of the Cabinet Secretariat of the Republic of Indonesia, the two Indonesian swimming athletes who competed at the Paris 2024 Olympic Games did not win any medals [5]. This condition indicates that the performance of Indonesian swimming athletes on the global stage still requires serious attention. Well planned and continuous athlete development and training can enhance the competitiveness of Indonesian swimmers at the international level [4].

The development of high performing swimming athletes requires a long-term and structured process, where coaches play a critical role in designing gradual and systematic training programs from an early age [6]. Swimming clubs serve as important institutions that facilitate athlete development and act as platforms for coaches to implement training strategies in preparation for competitions [7]. In this context, coaches are responsible for assessing athletes' physical characteristics and translating the assessment results into appropriate training programs [8].

In practice, the athlete development process is often disrupted by frequent changes in coaches and sports administrators within aquatic sports organizations. These transitions may lead to discontinuities in athlete development due to inadequate knowledge transfer between successive coaches and administrators. Important information such as training periodization, performance evaluation results, coaching assessments, and athlete development records is often not systematically documented in a structured knowledge management system [9]. As a result, coaching decisions are frequently based on incomplete historical data.

Moreover, several swimming coaches in Indonesia still rely heavily on intuition when assessing athletes physical characteristics and performance potential [10]. Intuition based assessments are highly dependent on personal experience and expertise, which may lead to subjective and inconsistent coaching decisions[11]. Such expertise and experience are not widely possessed by current coaches, as indicated by coaching licenses issued through training by the Indonesian Swimming Association (PRSI), which include A, B, and C licenses, relatively few coaches in Indonesia hold an A license to fill all swimming clubs [12][13]. Moreover, the absence of physical characteristic data may lead coaches to make less accurate decisions when designing training programs that suit swimmers needs [10][14] .

Physical characteristics play a crucial role in swimming performance. Interviews with the management of the South Sumatra Provincial Aquatic Association revealed that key physical attributes influencing swimming performance include height, body weight, arm length, and leg length. Body weight affects body size and buoyancy, which can enhance propulsive force in water [15]. Height and arm length contribute to reach and stroke efficiency, while leg length influences propulsion and movement speed [16]. In addition, age and sex affect physical development and lung capacity, which further impact swimming performance and buoyancy [1][17].

With the rapid advancement of technology, reliance solely on coaching intuition is no longer sufficient to support optimal athlete development. Technological tools, particularly those based on data analytics and machine learning, have increasingly been adopted in sport science to support objective performance evaluation and evidence based decision making [18][19]. Several previous studies have demonstrated the effectiveness of machine learning models in swimming analytics to predicting athletic performance based on physical characteristics. Rong et al. (2025) applied ensemble learning techniques to predict performance in sprinting, long distance running, and long jump using physical attributes such as age, sex, body weight, and lung capacity. Their results showed high prediction accuracy, with R^2 values reaching up to 0.97 [20]. Nurani et al. (2023) compared Decision Tree Regression and Multiple Linear Regression for modeling Body Mass Index (BMI) using demographic and health-related variables, where linear regression achieved superior performance with an R^2 of 0.9693 [21]. Kannanthara and Jose (2024) analyzed swimmers' performance using multiple machine learning models, including Lasso Regression, Random Forest Regression, and XGBoost, and found that Lasso Regression produced the best results with an average R^2 of 0.897 [22]. Furthermore, Nandedkar et al. (2024) utilized wearable sensor data to predict athlete performance and reported that Random Forest Regression achieved an R^2 of 0.9926. Chansanam et al. (2024) also evaluated various machine learning algorithms and found that XGBoost and Stacking Ensemble outperformed other models, achieving R^2 values above 0.99 [23].

Based on these five studies, R^2 is considered a key evaluation metric for measuring model accuracy, with values closer to 1 indicating lower prediction error and better model performance [24]. Among the evaluated methods, three models consistently demonstrated the highest R^2 values: XGBoost ($R^2 = 0.998465$), Random Forest ($R^2 = 0.992$), and Lasso Regression ($R^2 = 0.97$). Among these, XGBoost achieved the highest R^2 value and is widely recognized for its strong performance in handling tabular and non-linear data in both classification and regression tasks [25][26]. XGBoost also have incorporates randomization techniques and hyperparameter tuning mechanisms that help prevent overfitting, making it a robust and reliable model for predictive tasks [27]. So, XGBoost method will be used as a model in this study.

Despite these promising results, most existing studies primarily focus on model comparison or offline prediction accuracy and do not address the practical deployment of performance prediction models in real coaching environments. Moreover, limited studies have integrated machine learning-based performance prediction into a web-based system that translates prediction results into interpretable performance indicators suitable for coaching practices, particularly in Indonesia.

Therefore, this study aims to develop a web-based swimming performance assessment system that predicts race times based on swimmers' physical characteristics using an optimized XGBoost model. Unlike previous studies, this research not only evaluates model performance but also deploys the model within a fully functional website that presents prediction results in the form of estimated swimming time, performance percentage, and performance categories aligned with coaching standards. By providing an objective and data-driven tool, the proposed system is expected to reduce subjectivity in performance assessment, support coaching decision-making, and mitigate knowledge discontinuity caused by frequent changes in coaches.

II. Methods

The method used in this study is the Iterative Incremental Software Development Life Cycle (SDLC) method, the Iterative Incremental model is one of the SDLC models that combines iterative and incremental concepts in software development [28]. Iteration refers to a series of repeated processes in software development aimed at refinement or correction when errors are found, while increment refers to the process of building software by adding features that were not previously available [29][30]. Therefore, software development goes through a series of repetitions at each stage; once a stage is completed, an evaluation is conducted, and the results of this evaluation become the basis for development in the next iteration until the desired outcome is achieved [31]. These stages begin with the initiation phase, followed by iterative phases, and end with deployment, as illustrated in Figure 1. In the initiation phase, the initial development plan is determined, while the deployment phase involves implementing the software on a server so that it can be executed [32][33][34].

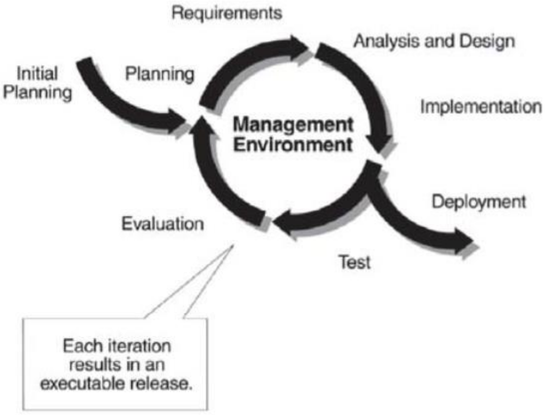


Figure 1. Iterative Incremental Method Cycle

Figure 1 illustrates this study flow, which based on the Iterative Increment model, which consists of five main stages: planning, analysis and design, implementation, testing, and evaluation. The Iterative Incremental method has advantages from both developer and user perspectives, as it is considered flexible for developing complex systems. This method allows developers to adapt the software to user needs and to quickly address issues that arise during development [33][35].

A. Planning

In the planning stage, dataset collection was conducted through manual measurements using measuring tapes and weighing scales on 94 respondents aged 6 to 14 years who participated in a fun swimming competition held from 23 to 24 November 2024 in Jakabaring, South Sumatra. The dataset consists of ten physical and race-related features, including age, sex, height, weight, left arm length, right arm length, total arm span length, leg length, swimming distance, and swimming stroke.

Furthermore, this stage also involved problem analysis based on relevant previous studies, as well as the preparation of a software development schedule, estimation of development costs, and validation of agreements regarding development costs, development timelines, and the dataset used, conducted in consultation with domain experts.

B. Analysis and design

The analysis and design stage determines the design of features to be developed based on the needs analysis of the identified problems in collaboration with experts. In this study, a prototype design is created using Figma before being implemented into the software. In addition to analyzing the most suitable model for this study, this stage involves a preliminary comparison of the selected XGBoost method with machine learning algorithms commonly used in previous studies, namely Random Forest and Lasso Regression. This comparison is conducted at an exploratory level to support the model selection process prior to system implementation.

C. Implementation

The implementation stage was carried out using Visual Studio Code as the primary code editor. The web application was developed using the Laravel PHP framework, which handles both backend and frontend functionalities, including user authentication, request handling, and user interface rendering. Laravel serves as the main application layer that manages user interactions and communicates with the prediction service.

The swimming performance prediction model was developed using the Python programming language and the XGBoost algorithm. Model development and experimentation were conducted in Google Colab, after which the trained model was deployed as a microservice using the Flask framework. This Flask-based microservice exposes an API endpoint that receives swimmer data from the Laravel application, performs data preprocessing and prediction, and returns the predicted swimming time.

The prediction microservice is hosted on the Hugging Face platform to ensure reliable deployment and accessibility. For data storage, the system utilizes Supabase as the database management system to store user account information, swimmer profiles, prediction results, and performance classification outcomes. The overall system architecture follows a client–server model with a microservice approach, enabling scalable integration between the web application and the prediction model.

D. Testing

The testing stage consists of two types of evaluation, functional testing and non-functional testing. Functional testing is conducted using the black-box testing approach, focusing on verifying system functionality without considering internal code structure. This testing includes evaluation of the prediction accuracy produced by the XGBoost model, verification of each system feature, and assessment of the user interface functionality. Non-functional testing is performed to evaluate software quality attributes, including usability, portability, and flexibility. These tests are conducted to ensure that the developed software meets the specified functional requirements and quality standards.

E. Evaluation

The evaluation stage is conducted on the developed software in collaboration with domain experts from the South Sumatra Provincial Aquatic Association and relevant stakeholders to ensure that all specified functional and non-functional requirements have been fulfilled. If any deficiencies or aspects requiring improvement are identified during the evaluation process, the development stages may be iteratively repeated to refine and enhance the system.

III. RESULT AND DISCUSSION

This section discusses the outcomes of the study based on three main components: the dataset, the prediction model, and the implemented software system. Unlike the methodology section, which describes the development process, this section focuses on explain the results obtained.

A. Dataset

The dataset used in this study was collected from 94 swimming athletes who participated in a fun swimming competition held in Jakabaring, South Sumatra. The athletes were aged between 6 and 14 years, representing beginner to junior level swimmers. This age range was selected to support early performance evaluation and talent identification in competitive swimming.

Although the data were collected from 94 individual athletes, each athlete contributed multiple records based on different swimming distances and stroke types. As a result, a total of 300 valid data instances were obtained after the preprocessing stage. This data structure allows the model to capture performance variations across multiple race conditions while maintaining consistency at the athlete level.

The input features used in the prediction model include age, sex, body height (cm), body weight (kg), left arm span length (cm), right arm span length (cm), total arm span (cm), leg length (cm), swimming distance (m), and swimming stroke type. These physical and race-related variables were selected due to their established relevance to swimming biomechanics and performance potential. The swimmer's name was recorded solely for identification and data management purposes and was not included as an input feature in the prediction model.

During the preprocessing phase, categorical encoding was applied to non-numerical variables to ensure compatibility with the XGBoost model. In addition, data records from athletes using assistive equipment such as freestyle kick boards, breaststroke kick boards, backstroke kick boards, and butterfly kick boards were excluded from the dataset. This exclusion was necessary because the Qualification Entry Time (QET) 2024 standards issued by the Indonesian Aquatics Federation do not provide benchmark times for athletes using assistive devices, which could otherwise introduce bias and reduce the validity of performance comparisons.

Model

Based on preliminary experiments, XGBoost achieved the best overall performance and was therefore selected for system deployment. The result are presented in Table 1.

TABLE I
MODEL PERFORMANCE COMPARISON

Model	Best Parameters	Metrics

Random Forest	'max_depth': None;
	'min_samples_leaf': 2;
	'min_samples_split': 2;
	'n_estimators': 100;
		R^2 = 0.8792708674;
		MSE = 317.9759473645;
		RMSE = 17.8318800850;
		MAE = 10.6370957584;

Lasso Regression	alpha = 0.00001	R^2 = 0.8584264619;
		MSE = 372.8758662678;
		RMSE = 19.3099939479;
		MAE = 13.0328722975;

XGBoost	**'colsample_bytree': 0.7;**
	'learning_rate': 0.1;
	'max_depth': 3;
	'n_estimators': 150;
	'subsample': 0.8;
		R^2 = 0.8959908226;
		MSE = 273.9389906213;
		RMSE = 16.5511023990;
		MAE = 9.1760189311;

Based on the evaluation metrics, XGBoost outperformed Random Forest and Lasso Regression in terms of R^2, MSE, RMSE, and MAE. These results indicate that XGBoost provides more accurate and consistent predictions compared to the other models. Following this selection, the overall process of model development and system implementation is described through a flowchart, as illustrated in Figure 2.

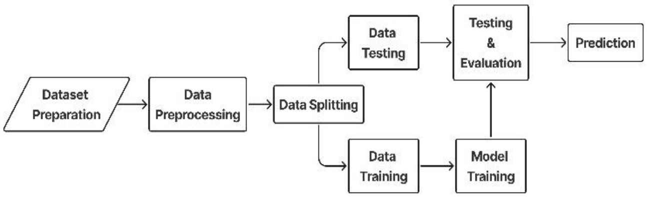


Figure 2. Model Development Flowchart

Based on the flowchart that illustrated in Figure 2, model application phase started when the dataset was ready to divided into 80 % training data and 20% testing data to be used as input for the model. In this study, GridSearchCV is employed to automatically determine the optimal hyperparameters for the XGBoost prediction model. GridSearchCV is a hyperparameter tuning method that performs an exhaustive evaluation of hyperparameter combinations to identify the most suitable configuration for model training [36]. Hyperparameter tuning plays a crucial role in improving the performance of prediction models [37].

The hyperparameters tuned in this study include *learning_rate*, *max_depth*, *n_estimators*, *gamma*, *reg_alpha*, *reg_lambda*, *colsample_bytree*, and *subsample*. Using the GridSearchCV approach, these hyperparameter combinations are evaluated through the K-Fold Cross-Validation method with *k = 5*, as illustrated in Figure 3.

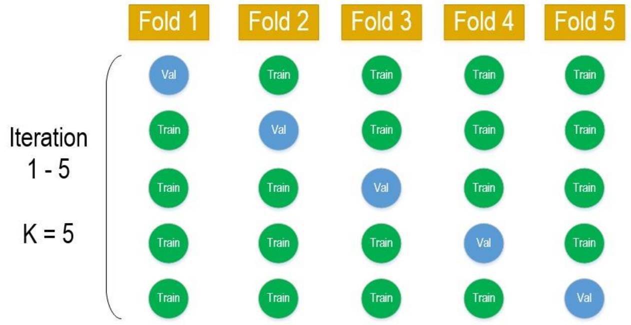


Figure 3. Illustration of The Grid Search Cross Validation Method

K-Fold Cross-Validation evaluates model performance by partitioning the dataset into *k* subsets, where each subset is used once as validation data while the remaining subsets are used for training. This process continues until all folds have been evaluated, allowing the model to achieve optimal performance. After the hyperparameter search is completed, the XGBoost model is trained using the best hyperparameter configuration identified during the tuning process. The trained model is then tested on the testing dataset to evaluate its predictive performance.

Model evaluation is conducted using the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-Squared (R^2) metrics on the validation results. The evaluation outcomes are subsequently visualized to examine the differences between the predicted values and the actual values. In this study, the model is evaluated by comparing cross-validation values of *cv = 5* and *cv = 10*. These values are selected because they are among the most commonly used settings in machine learning evaluation [38]. In addition, the evaluation results obtained from the original (non-normalized) data are compared with those from normalized data. The normalization technique applied in this study is *StandardScaler*, as it is a suitable preprocessing method for data standardization [39]. The hyperparameter combinations explored in this study are presented in Table 2.

TABLE II
BEST HYPERPARAMETER SEARCH COMBINATION

***Hyperparameter* (GridSearchCV)**	
max_depth	2, 3, 4, 5
learning_rate	0.01, 0.05, 0.1, 0.5
n_estimators	100, 200, 500, 1000, 2000
colsample_bytree	0.6, 0.7, 0.8, 0.9, 1
subsample	0.6, 0.65, 0.7

The hyperparameter combinations listed in Table 1 create 1,200 scenarios, with a total of 6,000 model fittings for *cv = 5* and 12,000 fittings for *cv = 10*, all of which are executed using Google Colab. The GridSearchCV result tables use abbreviations for each column name, including *LR* for *learning_rate*, *MD* for *max_depth*, *NE* for *n_estimators*, *CST* for *colsample_bytree*, and *SS* for *subsample*. A detailed comparison of these results is provided in Table 3.

TABLE III
GRIDSEARCHCV RESULTS COMPARISON

Data	Best Parameters	Rank	Metrics

Normal (cv = 5)	LR = 0.1;
	MD = 2;
	NE = 1000;
	CST = 0.8;
	SS = 0.65;
		1	R^2 = 0.95165580;
			MSE = 222.54094400;
			RMSE = 14.91780627;
			MAE = 8.76623429;

Normal (cv = 10)	**LR = 0.05;**
	MD = 2;
	NE = 1000;
	CST = 0.9;
	SS = 0.7;
		1	**R^2 = 0.95344497;**
			MSE = 213.28737255;
			RMSE = 14.60436142;
			MAE = 8.89788900;

Standard Scaler (cv = 5)	LR = 0.1;
	MD = 2;
	NE = 1000;
	CST = 0.8;
	SS = 0.65;
		1	R^2 = 0.95165580;
			MSE = 222.54094399;
			RMSE = 14.91780627;
			MAE = 8.76623428;

Standard Scaler (cv = 10)	LR = 0.05;
	MD = 2;
	NE = 2000;
	CST = 0.9;
	SS = 0.7;
		1	R^2 = 0.95344497;
			MSE = 213.28737254;
			RMSE = 14.60436142;
			MAE = 8.89788899;

That comparison results indicate that the use of StandardScaler does not have a significant impact on the model's performance. In contrast, applying *cv = 10* in dataset partitioning results in improved *R²* and *MAE* values, as well as reduced *MSE* and *RMSE* values, compared to *cv = 5*. The observed improvement in *R²* is 0.0018, while *MAE* improves by 0.1317. Meanwhile, *MSE* decreases by 9.2535 and *RMSE* decreases by 0.3134. Since *cv = 10* consistently provides better evaluation results than *cv = 5*, the final model training in this study adopts the non-normalized dataset with *cv = 10*. The evaluation results obtained using the optimal model are illustrated in Figure 4.

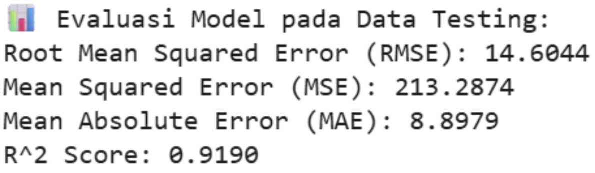

```
📊 Evaluasi Model pada Data Testing:
Root Mean Squared Error (RMSE): 14.6044
Mean Squared Error (MSE): 213.2874
Mean Absolute Error (MAE): 8.8979
R^2 Score: 0.9190
```

Figure 4. Testing Evaluation Results

To provide a comprehensive evaluation of the proposed model, multiple performance metrics were employed, including the coefficient of determination (R²) and error-based metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Based on the model testing results shown in Figure 5.22, an R² value of 0.9190 was obtained, indicating that the model explains approximately 92% of the variance in the data. The MAE value of 8.8979 indicates that the model's average prediction error is approximately 8.9 units from the actual values. Furthermore, the MSE value of 213.2874 and the RMSE value of 14.6044 indicate the presence of several prediction errors that are larger than the average error.

TABLE IV
R² INTERPRETATION

Score Range	Performance
0.8 - 1	Very High
0.6 – 0.79	High
0.4 – 0.59	Medium
0.2 – 0.39	Low
- ∞ – 0.19	Very Low

According to the R² interpretation categories presented in Table 4, these results indicate that the model demonstrates overall very high performance in predicting swimming time based on swimmers' physical characteristics. Therefore, the average prediction error (MAE) is considered relatively low, and the overall prediction error levels (MSE and RMSE) remain within an acceptable range for practical coaching applications.

B. Software

After evaluating the performance of the prediction model, this subsection discusses the implementation and testing results of the developed website, which was designed to deploy the proposed model in a practical environment. This software system architecture adopts a web-based client server model with a microservice approach to separate the prediction model from the main web application. This design improves modularity and allows independent development and deployment of the machine learning model. The main web application is responsible for managing user interactions, data input, and result visualization. The prediction model is implemented as a separate backend service that receives processed input data and returns prediction results. Communication between the web application and the prediction service is carried out through API-based requests, enabling efficient and scalable integration.

The technology used in backend and frontend of the web application are developed using the Laravel framework, which handles user authentication, request management, and interface rendering. The prediction model is implemented using the XGBoost algorithm and deployed as a microservice using the Flask framework based on Python. This microservice is hosted on the Hugging Face platform, allowing reliable model deployment and access through API endpoints. For data storage, the system utilizes Supabase as the database management system to store user accounts, swimmer data, prediction results, and performance classifications. This combination of technologies ensures a robust, scalable, and maintainable software architecture suitable for real world application. To provide a clearer understanding of the system operation, Figure 6 will presents the workflow of the proposed website in the form of a flowchart, which outlines the sequential process starting from user login and swimmer data input, followed by data preprocessing and prediction using the trained model, performance calculation, result visualization, and final storage of the outcomes in the database.

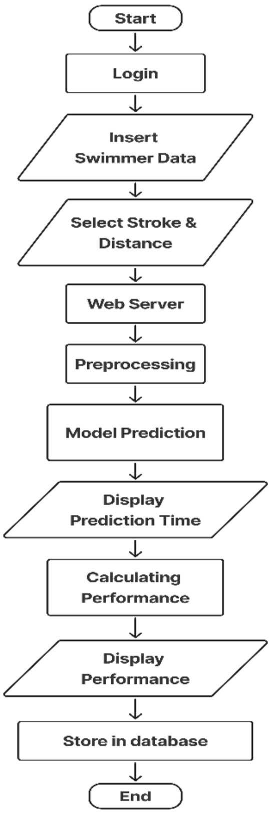


Figure 5. User Workflow

The process begins with user authentication, where coaches log in using accounts that have been prepared and verified by the system administrator. After successful authentication, the user is directed to the swimmer data input page, where swimmer profile data are entered and managed. The input at this stage includes the swimmer's name for identification purposes and physical characteristics, namely body height, body weight, left arm length, right arm length, total arm span length, and leg length. The swimmer's name is used solely for data management and record identification and is not included as an input feature in the prediction model. After the swimmer profile data are submitted, the data are stored in the system database and can be reused for prediction.

The user then proceeds to the prediction page, where swimming stroke and race distance are selected. These selections, together with the previously stored swimmer physical data, are sent by the frontend to the web server for further processing.

The backend of the web application is developed using the Laravel framework, which is responsible for handling user requests, validating input data, managing database operations, and communicating with external services. For prediction purposes, the trained XGBoost model is deployed as an independent microservice using the Flask framework based on Python and hosted on the Hugging Face platform. This microservice provides an API endpoint that receives the complete swimmer data from the Laravel backend.

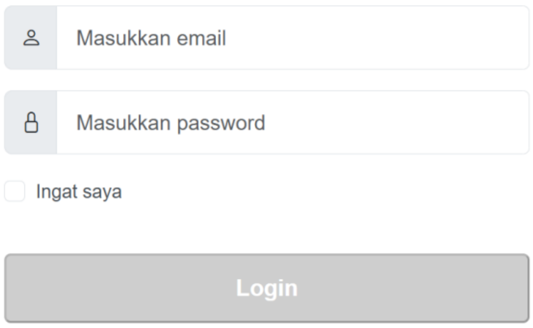
Before prediction is performed, the received data undergo preprocessing, including categorical encoding, to ensure compatibility with the input format required by the XGBoost model. The preprocessed data are then used by the model to generate a predicted swimming race time. Once the prediction process is completed, the predicted time is returned to the Laravel backend. The system subsequently calculates the swimmer's performance value using Equation (1) by comparing the predicted time with the applicable Qualification Entry Time (QET).

$$\frac{QET}{Prediction\ times}\ x\ 100\% \qquad (1)$$

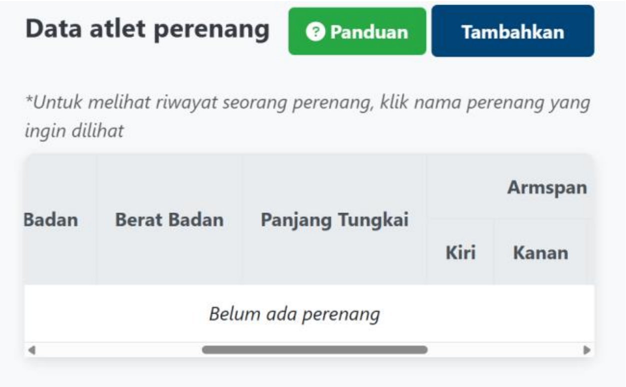
The obtained performance percentages are classified into four categories: Very High, High, Medium, and Low. The Very High category includes respondents whose performance percentages exceed 100%, indicating predicted swimming times faster than the Qualification Entry Time (QET). The High category consists of respondents with performance percentages ranging from 90% to 100%, the Medium category includes respondents with performance percentages between 60% and 90%, and the Low category includes respondents with performance percentages below 60%. These performance thresholds were determined based on official competitive swimming standards, specifically the QET 2024 benchmarks issued by Indonesian Aquatics Federation (Pengurus Besar Akuatik Indonesia), and were validated through consultations with the South Sumatra Provincial Aquatic Association. Therefore, the performance categorization is standards-based rather than derived from data distribution, ensuring that each category reflects meaningful competitive performance levels. Athletes achieving predicted times faster than QET benchmarks (>100%) are classified as having Very High performance potential, indicating readiness for provincial or national level competitions. A detailed distribution of respondents across these performance categories is presented in Table 5, while the software interface results are illustrated in Figure 6.

TABLE V
PERFORMANCE CATEGORIZATION DETAILS

Performance Category	Percentance (%)
Very High	> 100
High	$90 \le x \le 100$
Medium	$60 \le x < 90$
Low	< 60



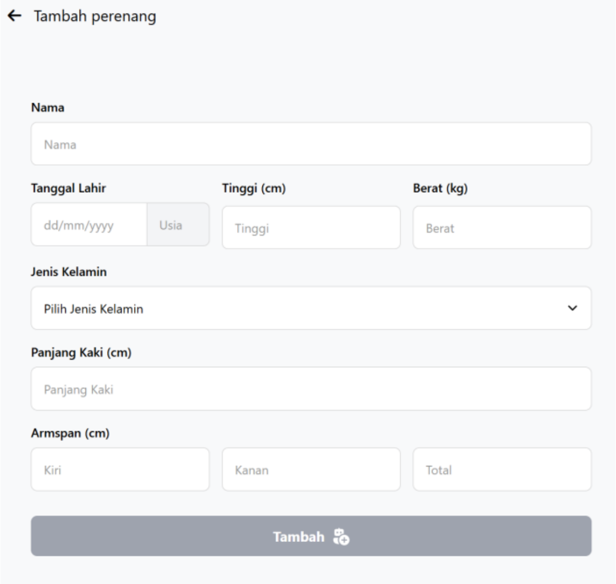
*Login Page Interface



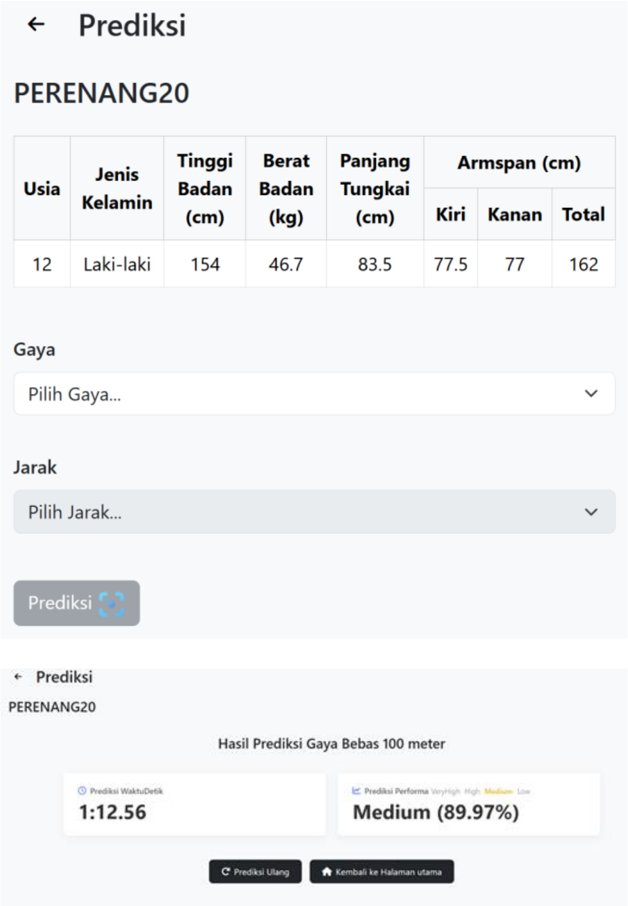
*Untuk melihat riwayat seorang perenang, klik nama perenang yang ingin dilihat

Badan	Berat Badan	Panjang Tungkai	Armspan	
			Kiri	Kanan
Belum ada perenang				

*Home Page Interface



*Add Swimmer Interface



← Prediksi

PERENANG20

Usia	Jenis Kelamin	Tinggi Badan (cm)	Berat Badan (kg)	Panjang Tungkai (cm)	Armspan (cm)		
					Kiri	Kanan	Total
12	Laki-laki	154	46.7	83.5	77.5	77	162

Gaya: Pilih Gaya...

Jarak: Pilih Jarak...

← Prediksi

PERENANG20

Hasil Prediksi Gaya Bebas 100 meter

Prediksi WaktuDetik: 1:12.56

Prediksi Performa VeryHigh High Medium Low: Medium (89.97%)

*Prediction Page Interface

Figure 6. Software Interface Result

The sofware testing phase will evaluates the system using the Black Box Testing approach, which is divided into two types: functional testing and non-functional testing. Functional testing is a software testing process in which test cases focus on how the software interacts with its functional services. The tested functions examine how the system operates under specific conditions, such as receiving input data and observing how the system processes the input and produces output [40]. In this study, the functional testing criteria using the Black Box Testing method are presented in Table 6.

TABLE VI
FUNCTIONAL TESTING CRITERIA

No	Testing Criteria
1	Users can input data into the system for prediction purposes.
2	The system displays the predicted swimming time, performance percentage, and performance category (*very high, high, medium,* or *low*).
3	Users can access the interface through a web-based application and operate all features without any issues.
4	The system is able to complete the swimming time prediction process in less than 10 seconds.

Non-functional testing is a testing process that describes the operational capabilities and limitations of software and aims to enhance system quality [41]. This type of testing indirectly evaluates how well the developed system performs in operation, particularly in terms of response time relative to predefined requirements [40]. A total of twelve non functional testing parameters are illustrated in Figure 7.

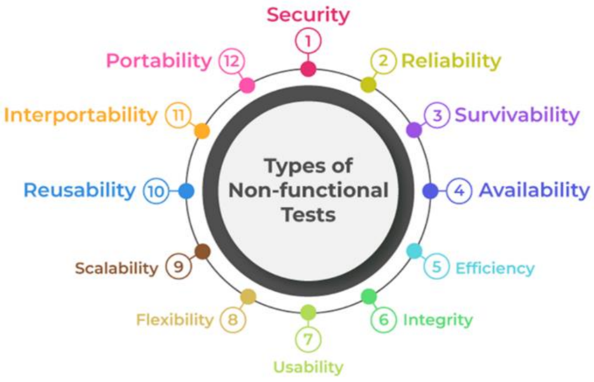


Figure 7. Non-Functional Testing Parameters

In this study, non-functional testing focuses on three out of the twelve parameters presented in Figure 7. The specific non-functional testing criteria applied in this study are shown in Table 7.

TABLE VII
NON FUNCTIONAL TESTING CRITERIA

No	Testing Criteria	Parameter
1	The software provides a user friendly interface.	*Usability*
2	The software can be accessed through devices that support web browsing.	*Portability*
3	The software is capable of predicting swimmers potential race times in less than 10 seconds.	*Flexibility*

Based on these criteria, each software feature was tested accordingly. In addition, response time testing was conducted ten times to determine the average time required for the system to perform predictions. The results of the response time testing are presented in Table 8.

TABLE VIII
TESTING PREDICTION TIME

Testing Number	Prediction Time (Seconds)
1	7.26
2	7.08
3	7.19
4	7.78
5	7.20
6	7.22
7	7.09
8	6.51
9	6.68
10	6.35
Average Time	**7.036**

Based on the testing results, it can be concluded that all Black Box Testing criteria were successfully met. The detailed results of the functional testing are summarized in Table 9.

TABLE IX
BLACK BOX TESTING RESULTS

No	Testing Criteria	Status
1	Users can input data into the system for prediction purposes.	Success
2	The system displays the predicted swimming time, performance percentage, and performance category (*very high, high, medium,* or *low*).	Success
3	Users can access the interface through a web-based application and operate all features without any issues.	Success
4	The system is able to complete the swimming time prediction process in less than 10 seconds.	Success

After completing functional testing using Black Box Testing, non-functional testing was conducted to evaluate application satisfaction through the USE questionnaire. The questionnaire was administered in paper form to one expert and nine stakeholders (coaches) after a demonstration of the application. The responses were assessed using a five point Likert scale, as shown in Table 10 [42].

TABLE X
LIKERT SCALE

Description	Score	Percentage (%)
Strongly Agree	5	80 – 100
Agree	4	60 – 79.99
Neutral	3	40 – 59.99
Disagree	2	20 – 39.99
Strongly Disagree	1	0 – 19.99

The Likert scale scores were calculated based on the questionnaire results using Equation (2). Subsequently, the percentage scores were computed using Equation (3) [43].

$$Total\ Score\ =\ T\ x\ Pn \qquad (2)$$

With:
T = Total number of responses for each scale
Pn = Likert scale value

$$\%\ = \frac{Total\ Score}{Highest\ Score\ x\ Total\ Answer}\ x\ 100\% \qquad (3)$$

With:
% = Likert scale percentage index

The questionnaire items were based on the USE Questionnaire method, which covers four main aspects: usefulness, ease of use, ease of learning, and satisfaction. The list of questions corresponding to these aspects is presented in Table 11, while the USE questionnaire results are shown in Table 12.

TABLE XI
LIST OF USE QUESTIONNAIRE

Category	No	Questions
Usefulness	1.	The application can be used effectively to predict swimmers performance.
	2.	The application provides initial prediction results regarding swimmers performance potential as preliminary information to assist coaches in making training decisions.
	3.	I find this application useful as a supporting tool in the development of young swimmers.
Ease of Use	4.	The application can be used without difficulty.
	5.	The application interface is responsive and user-friendly.
	6.	When an error occurs, the application displays error messages that are easy to understand.
Ease of Learning	7.	The operation of the application is easy to understand.
	8.	I can easily remember the steps required to use the application.
	9.	I am able to use this application without difficulty within a short period of time.
Satisfaction	10.	I am satisfied with the features and performance provided by the application.
	11.	The use of colors, icons, and images in the application is satisfactory.
	12.	Overall, I am satisfied with the application.

TABLE XII
USE QUESTIONNAIRE RESULTS

No	Question	Percentange	Description
1.	The application can be used effectively to predict swimmers performance.	86%	Strongly Agree
2.	The application provides initial prediction results regarding swimmers performance potential as preliminary information to assist coaches in making training decisions.	86%	Strongly Agree
3.	I find this application useful as a supporting tool in the development of young swimmers.	84%	Strongly Agree
4.	The application can be used without difficulty.	90%	Strongly Agree
5.	The application interface is responsive and user-friendly.	88%	Strongly Agree
6.	When an error occurs, the application displays error messages that are easy to understand.	88%	Strongly Agree
7.	The operation of the application is easy to understand.	94%	Strongly Agree

8.	I can easily remember the steps required to use the application.	88%	Strongly Agree
9.	I am able to use this application without difficulty within a short period of time.	88%	Strongly Agree
10.	I am satisfied with the features and performance provided by the application.	88%	Strongly Agree
11.	The use of colors, icons, and images in the application is satisfactory.	86%	Strongly Agree
12.	Overall, I am satisfied with the application.	92%	Strongly Agree

Based on the questionnaire results obtained using the USE method, the developed application was overall classified in the *strongly agree* category, achieving an average satisfaction percentage of 88.16%.

IV. Conclusion

Based on the testing results and the developed website based swimming performance potential prediction software, several conclusions can be drawn as follows:

1. The developed software successfully predicts swimming time to assess swimmers performance potential based on age, sex, and physical characteristics.
2. The software was successfully implemented functionally on a website platform with key features including user authentication, swimmer data management, performance prediction, and prediction history.
3. The software met the specified requirements for predicting swimmers performance potential, achieving an average prediction processing time of 7.036 seconds.
4. The prediction model utilized in the software employed the best-performing model obtained from training with cv = 10. The model testing results demonstrated very high performance, achieving an R^2 value of 0.9190, *MSE* of 213.2874, *RMSE* of 14.6044, and *MAE* of 8.8979.
5. The questionnaire results obtained using the USE Questionnaire method indicate that the developed software falls into the *strongly agree* category, with an average satisfaction percentage of 88.16%.
6. A limitation of this study is the absence of longitudinal validation comparing predicted performance with actual competition results. Future work should conduct field testing to validate predictions against real race times over multiple competitions.

References

[1] I. M. S. Mahardika, *Renang the science of swimming*. Jawa Timur: Zifatama Jawara, 2024.

[2] Martinus, Tandiyo Rahayu, Rumini, and Tri Rustiadi, *Pembinaan prestasi cabang olahraga renang*, 1st ed., vol. 1. Jawa Tengah: Zahira Media Publisher, 2021.

[3] E. Rahayu and O. P. Mulyana, "Hubungan antara goal-setting dan motivasi berprestasi dengan prestasi atlet renang," *Character: Jurnal Penelitian Psikologi*, vol. 3, no. 3, Feb. 2015, doi: 10.26740/cjpp.v3i3.10947.

[4] D. Setyaningsih, A. Soemarmi, and U. S. Hardjanto, "Pelaksanaan Peraturan Presiden Nomor 95 Tahun 2017 tentang peningkatan prestasi olahraga nasional pada olahraga akuatik," *Diponegoro Law Journal*, vol. 8, no. 1, pp. 184–192, Jan. 2019, doi: 10.14710/dlj.2019.25333.

[5] Sekretariat Kabinet Republik Indonesia, "Klasemen akhir Olimpiade Paris 2024: Amerika Serikat posisi pertama, Indonesia peringkat ke-39." Accessed: May 14, 2025. [Online]. Available: https://setkab.go.id/klasemen-akhir-olimpiade-paris-2024-amerika-serikat-posisi-pertama-indonesia-peringkat-ke-39/

[6] M. S. Anjasmoro and T. Soenyoto, "Survei pembinaan prestasi atlet renang di Pemalang: Studi pada Shark Swimming Club tahun 2021," *Indonesian Journal for Physical Education and Sport*, vol. 4, no. 1, Jul. 2023, doi: 10.15294/inapes.v4i1.51894.

[7] S. Susilawati, "Implementasi manajemen pelatih renang di Club Ciamis Aquatic," *Indonesian Journal of Education Management and Administration Review*, vol. 4, no. 1, pp. 213–219, Jun. 2020.

[8] S. Sugito, M. A. H. Allsabah, and R. P. Putra, "Manajemen kepelatihan klub renang Kota Kediri tahun 2019," *Jurnal SPORTIF: Jurnal Penelitian Pembelajaran*, vol. 6, no. 1, pp. 1–271, Apr. 2020, doi: 10.29407/js_unpgri.v6i1.14021.

[9] A. Widi, E. Sediyono, K. D. Hartomo, I. Sembiring, and A. Iriani, "Development of Knowledge Management System with Soft System Metodhology in Aquatic Organization," in *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, Oct. 2022, pp. 143–146. doi: 10.23919/EECSI56542.2022.9946489.

[10] S. Sugiarto and A. B. Hananto, "Kontribusi karakteristik antropometri terhadap kemampuan fisik dan kinerja renang gaya bebas," *Prosiding Seminar Nasional Olahraga*, vol. 4, no. 1, pp. 23–31, Jun. 2022.

[11] P. Alexiou *et al.*, "A descriptive study of coaches' perceptions regarding talent identification and development in swimming," *International Journal of Human Movement and Sports Sciences*, vol. 12, no. 2, pp. 326–336, Feb. 2024, doi: 10.13189/saj.2024.120207.

[12] J. Juriana, K. Tahki, and G. Zulfitranto, "Pengetahuan psikologi olahraga pada pelatih renang Indonesia," *Jurnal ilmiah Sport Coaching and Education*, vol. 2, no. 1, pp. 31–38, Jan. 2018.

[13] F. B. Negara, Y. Yarmani, and Y. E. Nopiyanto, "Pengetahuan psikologi olahraga pada pelatih renang dengan prestasi atlet renang di Rejang Lebong," *Sport Gymnatics: Jurnal Ilmiah Pendidikan Jasmani*, vol. 2, no. 2, pp. 228–239, Oct. 2021, doi: 10.33369/gymnastics.

[14] S. Ganzevles, R. Vullings, P. Jan Beek, H. Daanen, and M. Truijens, "Using tri-axial accelerometry in daily elite swim training practice," *Sensors*, vol. 17, no. 5, pp. 1–14, Apr. 2017, doi: 10.3390/s17050990.

[15] Q. F. Hidayat, A. W. B. Utomo, and A. S. Prayoga, "Hubungan antara tinggi badan, panjang tungkai, dan kekuatan otot lengan terhadap prestasi renang gaya dada 50 meter pada perenang Tirta Lawu Swimming Club usia 13–14 tahun di Kecamatan Ngrambe, Kabupaten Ngawi," *Journal Sport Science Indonesia (JASSI)*, vol. 3, no. 2, pp. 400–409, May 2024, doi: 10.31258/jassi.3.2.415-429.

[16] R. H. Ali, G. F. Yosika, and D. Pranata, "Identifikasi profil antropometri atlet klub renang di Kota Pontianak," *Jurnal Pendidikan Jasmani Khatulistiwa*, vol. 3, no. 2, pp. 149–156, Nov. 2022, doi: 10.26418/jpjk.v3i2.64035.

[17] W. Prasiwi and S. Darnoto, "Hubungan antara usia dan masa kerja dengan kapasitas fungsi paru pada supeltas di Surakarta,"

Prosiding Semnas & Call for Papers Prodi Kesehatan Masyarakat 2017, pp. 68–71, Jul. 2017.

[18] A. Rohendi and H. Rustiawan, “Kebutuhan sport science pada bidang olahraga prestasi,” *Research Physical Education and Sports (Journal RESPECS)*, vol. 2, no. 1, pp. 32–43, Jan. 2020, doi: 10.31949/jr.v2i1.2013.

[19] A. Y. Kurniawan, M. Rozaq, and A. Diana, “Penggunaan teknologi digital dalam pembelajaran sains dan olahraga untuk meningkatkan literasi dan pemahaman siswa,” *Journal Sport Science, Health and Tourism of Mandalika (Jontak)*, vol. 6, no. 1, pp. 33–42, Jan. 2024, doi: 10.36312/jontak.v6i1.4053.

[20] H. Rong, Q. Zewen, M. Hao, H. Zhezhe, and X. Yue, “Sports performance prediction for college students through ensemble learning algorithm,” *IEICE Trans Inf Syst*, pp. 1–8, Jan. 2025.

[21] A. T. Nurani, A. Setiawan, and B. Susanto, “Perbandingan kinerja regresi Decision Tree dan regresi linear berganda untuk prediksi BMI pada dataset Asthma,” *Jurnal Sains dan Edukasi Sains*, vol. 6, no. 1, pp. 34–43, Feb. 2023, doi: 10.24246/juses.v6i1p34-43.

[22] A. Kannanthara and M. Jose, “Optimizing performance analysis in swimming races: Exploring segmented approaches to race behaviour,” *Digitala Vetenskapliga Arkivet (DiVA)*, pp. 1–33, Jun. 2024.

[23] A. Nandedkar, C. Gadve, C. Gowda, S. Deep, and R. Mulla, “Athlete performance prediction using Random Forest,” *Int J Res Appl Sci Eng Technol*, vol. 12, no. 5, pp. 2370–2377, May 2024, doi: 10.22214/ijraset.2024.62112.

[24] W. Chansanam *et al.*, “Enhancing predictive accuracy in educational assessment: A comparative analysis of machine learning models for predicting student performance,” *Review of Contemporary Philosophy*, vol. 23, no. 1, pp. 142–160, Jun. 2024.

[25] J. Brownlee, *XGBoost with Python: Gradient boosted trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2016.

[26] I. Maulita and A. M. Wahid, “Prediksi magnitudo gempa menggunakan Random Forest, Support Vector Regression, XGBoost, LightGBM, dan Multi-Layer Perceptron berdasarkan data kedalaman dan geolokasi,” *Jurnal Pendidikan dan Teknologi Indonesia (JPTI)*, vol. 4, no. 5, pp. 221–232, Dec. 2024, doi: 10.52436/1.jpti.470.

[27] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2020, doi: 10.1007/s10462-020-09896-5.

[28] Y. Rahmadi, Y. A. P, and M. A. H, “Pengembangan modul freemium aplikasi TEL-US (Telkom University Store) menggunakan metode iterative incremental dan framework Laravel,” *e-Proceeding of Engineering*, vol. 2, no. 2, pp. 5437–5444, Aug. 2015.

[29] S. E. Prastya, M. C. Saputra, and D. Pramono, “Pengembangan sistem informasi data pasien Seksi Rehabilitasi BNN Kota Malang menggunakan metode iterative incremental,” *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 2, no. 12, pp. 6587–6596, Dec. 2018.

[30] D. H. Syahputra, R. Hanafi, and R. Andreswari, “Perancangan sistem informasi pemasaran Invenit pada PT. Sibayak Gugung Nari berbasis website menggunakan metode iterative incremental,” *e-Proceeding of Engineering*, vol. 8, no. 1, pp. 634–647, Feb. 2021.

[31] R. S. Afifah, F. M. Al Anshary, and M. Hardiyanti, “Pengembangan aplikasi roll bagi customer berbasis website menggunakan metode iterative dan incremental,” *eProceedings of Engineering*, vol. 9, no. 2, pp. 650–655, Apr. 2022.

[32] I. A. Faruqi, S. F. S. Gumilang, and M. A. Hasibuan, “Perancangan back-end aplikasi Rumantara dengan gaya arsitektur REST menggunakan metode iterative incremental,” *e-Proceeding of Engineering*, vol. 5, no. 1, pp. 1411–1417, Mar. 2018.

[33] K. D. Permana, R. Fauzi, and S. Suakanto, “Pengembangan backend investasi berbasis website pada ekosistem digital Ihya dengan metode iterative incremental,” *Jurnal Riset Komputer (JURIKOM)*, vol. 9, no. 5, pp. 1226–1233, Oct. 2022, doi: 10.30865/jurikom.v9i5.4830.

[34] A. Zulfa, A. Kusyono, T. N. Adi, and E. L. Thohiroh, “Pengembangan website edukasi kesehatan balita dengan menggunakan metode iterative incremental,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 5, no. 1, pp. 263–274, Aug. 2024, doi: 10.30865/klik.v5i1.1962.

[35] Y. I. Chandra, M. Riastuti, and D. Kosdiana, “Penerapan model iterative incremental dalam membangun aplikasi e-commerce di Toko Kopi Rarukuh Luas berbasis web mobile,” *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, vol. 7, no. 1, pp. 179–190, Jul. 2023.

[36] I. Muhamad and M. Matin, “Hyperparameter tuning menggunakan GridsearchCV pada Random Forest untuk deteksi malware,” *Journal Multimedia Networking Informatics (MULTINETICS)*, vol. 9, no. 1, pp. 43–50, May 2023, doi: 10.32722/multinetics.v9i1.5578.

[37] W. Nugraha and A. Sasongko, “Hyperparameter tuning on classification algorithms using Grid Search,” *SISTEMASI: Jurnal Sistem Informasi*, vol. 11, no. 2, pp. 391–401, May 2022, doi: 10.32520/stmsi.v11i2.1750.

[38] I. K. Nti, O. Nyarko-Boateng, and J. Aning, “Performance of machine learning algorithms with different K values in K-fold cross-validation,” *Article in International Journal of Information Technology and Computer Science*, vol. 6, pp. 61–71, Dec. 2021, doi: 10.5815/ijitcs.2021.06.05.

[39] A. B. Majumder *et al.*, “Heart disease prediction using concatenated hybrid ensemble classifiers,” *Algorithms*, vol. 16, no. 12, Dec. 2023, doi: 10.3390/a16120538.

[40] R. A. Sianturi, A. M. Sinaga, Y. Pratama, H. Simatupang, J. Panjaitan, and S. Sihotang, “Perancangan pengujian fungsional dan nonfungsional aplikasi SIAPPARA di Kabupaten Humbang Hasundutan,” *Jurnal Komputer dan Informatika (J-ICON)*, vol. 9, no. 2, pp. 133–141, Sep. 2021, doi: 10.35508/jicon.v9i2.4706.

[41] S. F. Muthmainnah and H. P. Putro, “Pengujian nonfungsional dengan pendekatan McCall’s Factor pada perspektif product revision,” *AUTOMATA*, vol. 4, pp. 1–8, Nov. 2023.

[42] V. H. Pranatawijaya, W. Widiatry, R. Priskila, and P. B. A. A. Putra, “Penerapan Skala Likert dan Skala Dikotomi Pada Kuesioner Online,” *Jurnal Sains dan Informatika*, vol. 5, no. 2, pp. 128–137, Dec. 2019, doi: 10.34128/jsi.v5i2.185.

[43] M. Fiqih Erinsyah, G. W. Sasmito, D. S. Wibowo, and V. K. Bakti, “Sistem evaluasi pada aplikasi akademik menggunakan metode skala Likert dan algoritma Naïve Bayes,” *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 13, no. 1, pp. 74–82, Apr. 2024, doi: 10.34010/komputa.v13i1.10940.