



ISSN: 0067-2904

## Implementation of K-Nearest Neighbors Algorithm for Predicting Heart Disease Using Python Flask

Dimas Aryo Anggoro\*, Nur Chudlori Aziz

Informatics Department, Universitas Muhammadiyah Surakarta, Indonesia

Received: 13/10/2020

Accepted: 15/12/2020

### Abstract

Heart disease is a non-communicable disease and the number 1 cause of death in Indonesia. According to WHO predictions, heart disease will cause 11 million deaths in 2020. Bad lifestyle and unhealthy consumption patterns of modern society are the causes of this disease experienced by many people. Lack of knowledge about heart conditions and the potential dangers cause heart disease attacks before any preventive measures are taken. This study aims to produce a system for Predicting Heart Disease, which benefits to prevent and reduce the number of deaths caused by heart disease. The use of technology in the health sector has been widely practiced in various places and one of the advanced technologies is machine learning. Machine learning technology can be used to predict the potential patients of heart disease by implementing the K-Nearest Neighbors (KNN). The algorithm results in 65.93% for its accuracy, which is then improved to 82.41% due to the z-score normalization. It shows that z-score can noticeably improve the accuracy of the KNN algorithm. The system is developed based on a website that uses the Flask micro-framework so that development is more efficient. Flask is a micro-framework based on the Python programming language that does not contain many tools and libraries, so it is more portable and does not utilize a lot of resources.

**Keywords:** Heart Disease, Machine Learning, K-Nearest Neighbors, Flask, Python.

### 1. Introduction

As time goes by, access to many things becomes easier without having to spend a lot of energy. Numerous types of food and beverages develop in various ways but some tend to harm the human body due to the high fat and calorie, but low in fiber content [1]. It is undeniable that in this era of acceleration, human beings cannot waste their time merely waiting for food and will look for fast food or junk food, which is served as the main food, even though it tends to contain low nutritional value [2]. Non-communicable diseases (NCD), which are the result of an unhealthy lifestyle, are the cause of 36 million deaths or 63% of total deaths occurring in the world every year [3].

One example of the NCDs is the heart disease, which is the number 1 cause of death in Indonesia [4]. Heart disease is a concern for many people because of its sudden onset and can cause sudden death. There are several types of heart disease, including coronary heart disease and arrhythmias. Coronary heart disease (CHD), according to the World Health Organization (WHO) in 2002, caused 7 million death cases worldwide and is predicted to reach 11 million in 2020 [5]. Coronary heart disease occurs when there is a build-up of plaque in the coronary arteries so that the oxygen supply to the heart is not optimal [6]. Arrhythmias as a disease that is not widely known by the public, but in 2011 there were 2.1 million cases recorded [7]. Arrhythmias is a disorder of the heart that occurs due to the abnormal propagation of electrical impulses in the myocardium [8]. It causes a slow, fast, or irregular heartbeat. It is necessary to have knowledge of the condition of an individual's body, which can be achieved using technology. Since the 20th century, technology has become a popular medium in research as a

\*Email: dimas.a.anggoro@ums.ac.id

solution to many problems [9], including in the health sector. The use of technology to predict a person suffering from heart disease will be more effective using clinical data compiled from cases that have occurred. The growth of technology is at its peak with machine learning technology which has become a part of artificial intelligence [10].

Machine learning has an important role in improving the quality of health services because it can present medical diagnoses tools to predict diseases [10]. One algorithm that can be used for prediction is, which is applied for classification and is included in the supervised machine learning method [11]. The KNN algorithm works by finding the closest attributes from the training data and classifying them [12]. This algorithm has several advantages; for instance, it does not consume time for the training process, is effective with large training data, and is simple to learn [13]. In a previous study, Andiani [14] compared the KNN algorithm with the Random Forest algorithm in heart disease analysis and produced KNN with a better accuracy rate of 0.93% compared to Random Forest which was only 0.73 %.

This study aims to implement the KNN algorithm in predicting the potential for heart disease using the Python Flask. Flask is a micro-framework written in the Python programming language which does not contain many tools and libraries [15]. Flask is included in the micro-framework so that the resources used are not large and can be allocated to the KNN process. Hence, Flask is precisely used for implementing the KNN algorithm.

Some studies on the use of KNN regarding heart disease only focus on analyzing and finding the accuracy of the algorithm. In this study, a web-based system was built that anyone who works with health data can use to determine the potential for heart disease so that prevention can be achieved early on.

The findings of this study can be used to prevent and reduce the number of deaths caused by heart disease. These tasks are not handled properly due to the patient's unconsciousness and lack of knowledge about the patient's heart condition. The wider community, particularly health workers, can use the results of this research to support learning related to heart disease.

## 2. Methods

### 2.1. Data Collection

Data were obtained from the University of California, Irvine (UCI) Machine Learning Repository which consists of 4 databases, namely Cleveland, Hungary, Switzerland, and Long Beach VA [14]. This dataset has 303 rows of data and 13 attributes with one additional attribute called 'target' which has 2 class labels: heart disease (1) and non-heart disease (2) [16]. There were 138 data of non-heart disease and 165 data of heart disease. The following is a detailed description of each attribute.

**Table 1-**Description of heart disease data attributes from UCI Machine Learning Repository

Attribute	Information
Age	Age
Sex	Sex
Cp	Chest Pain Type. This attribute contains 4 aspects: typical angina, atypical angina, non-anginal pain, and asymptomatic
Trestbps	Patient's blood pressure at rest period or no activity
Chol	Cholesterol level
Fbs	Blood sugar level, if exceeds 120 mg/dl it has True value and less than 120 mg/dl has False value
Restecg	Resting Electrocardiographic Results, Electrocardiographic test results while the patient is at rest. With values: 0 for normal, 1 has ST-T wave abnormalities, and 2 indicates a definite probability of left ventricular hypertrophy based on Estes criteria
Thalach	Maximum heart rate
Exang	Pain experienced by the patient while exercising. True value if 'yes' and False if 'no'

Oldpeak	The decrease in ST due to exercise
Slope	Slope at peak ST after exercising. It has 3 types of values: upsloping, flat, downsloping
Ca	The number of blood vessels detected through the staining process
Thal	Thalassemia examination. Contains 3 values: normal, fixed defect, reversible defect
Target	Class label: heart disease (1) non-heart disease (0)

**2.2. Data Preprocessing**

The data used as training data is not consistently ideal. Thus, the raw data needs to be processed in a method called data preprocessing. Data preprocessing functions to improve data formats and clean up any disturbances or noise in the raw data [17]. In this study, two kinds of data preprocessing methods were used:

**A. Z-score Normalization**

Before applying Principal Component Analysis (PCA), the data to be processed needs to be normalized using the Z-score Normalization method. This method works based on the mean and standard deviation of the data, so that the specified range can be obtained from the dataset. In the research conducted by Anggoro & Supriyanti [18], it was found that this method can increase the accuracy of the model. The formula for Z-score Normalization is shown in equation 1.

$$X_i' = \frac{x_i - \mu}{\sigma} \tag{1}$$

where  $X_i'$  = Normalized data;  $x_i$  = Original data;  $\mu$  = Average of data; and  $\sigma$  = Standard deviation of data.

**B. Principal Component Analysis**

To avoid curse of dimensionality due to many variables, it is necessary to reduce the variable to several variables. Dimensionality reduction is a process carried out to reduce a large number of variables to be leaner and more efficient. This study performed dimensionality reduction using the PCA method. PCA is widely used in high-dimensional data [19], as in this study which used data with 13 dimensions. PCA reduces the number of interconnected dimensions to a collection of smaller dimensions called a principal component [20]. There will be no more principal components than the initial data dimensions [21]. By extracting features using eigenvectors and eigenvalues, the complexity of the data dimension can be reduced [22]. Jolliffe [23] revealed that PCA is a less dominant factor without changing the meaning of the original data. According to Adiwijaya [22], steps to reduce dimensions using PCA are as follows:

X is initiated with training data consisting of n-vectors with data dimensions (m). Calculating the average of each dimension can be conducted using equation 2.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \tag{2}$$

where n is the amount of data and  $X_i$  is the observation data. Then, the covariance matrix ( $C_x$ ) is calculated using equation 3.

$$C_x = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T \tag{3}$$

where  $\bar{X}$  is the mean of the data. Determining eigenvalues and eigenvectors is performed by using equation 4.

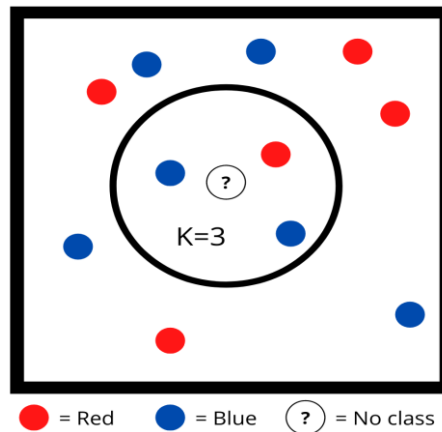
$$C_x v_m = \lambda_m v_m \tag{4}$$

After finding the eigenvalues, they are sorted from the largest to the smallest. The set of eigenvectors corresponding to the sorted eigenvalues will become the principal component. The principal component dimensions will be reduced based on the eigenvalues.

**2.3. Data Processing**

**1. K-Nearest Neighbors**

KNN is a supervised algorithm that classifies objects in which class is not yet determined and finds out the closest objects with the same class. The number of closest objects is determined based on the value of K. The value of K cannot be assigned with 1 or an even value.



**Figure 1-** Plot of data using KNN algorithm

In Figure- 1, it is explained that there are two classes or labels, with the use of the value of  $K = 3$ . There are 3 objects closest to the object that does not have class, which are 2 blue objects and 1 red object. Thus, objects that do not have class will have a blue class since the closest object to the value of  $K = 3$  is the blue class.

To find objects with the closest distance, the Euclidean distance is used as shown in equation 5.

$$D_i = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (5)$$

where  $D_i$  = Distance of the  $i$ -th variable;  $i$  = data variable ( $i = 1, 2, 3, \dots, n$ );  $n$  = data dimension;  $p_i$  = testing data,  $q_i$  = training data.

First, the value of  $K$  is determined. Then, the distance between the testing data and the training data is calculated using the formula in equation 5. The distance from the closest to the farthest distance is sorted. The object with the closest distance is taken as much as  $K$  values. The most targets are determined based on the closest one and then the class is assigned to the tested object.

According to Amra [24], KNN has several advantages. This algorithm is easy to understand and implement, quick in training data, suitable for data with a lot of noise, and proper for data with many classes. However, in addition to these advantages, KNN has several disadvantages, such as being a lazy learning algorithm, rather slow, and sensitive to local data structures, as well as consuming a lot of resources, especially memory,

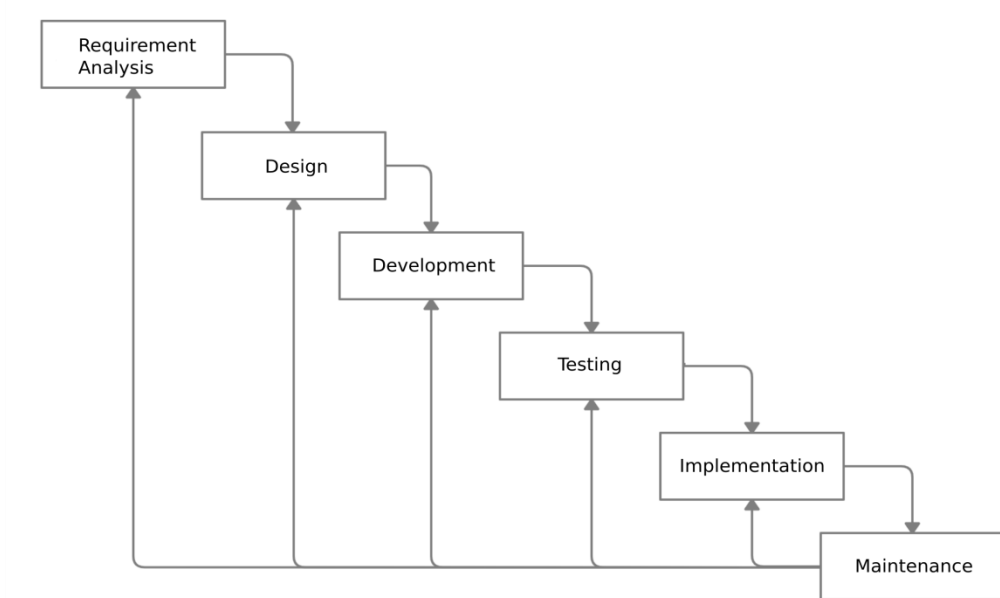
#### 2.4. Model Evaluation

To find out the accuracy of the model from the algorithm, model evaluation is required. This paper used the Cross-Validation method, which is a statistical method used to evaluate learning performance and predictive performance on unknown datasets [25]. In Cross-Validation, the dataset is divided into two parts, namely the training data and testing data. Each datum will be part of the training data or testing data with this method [26]. Cross-Validation has the form of  $k$ -fold validation, in which the value of  $k$  will determine the data distribution; for instance, if using 8-fold validation, the dataset is divided into 7 subsets for training sets and 1 for testing sets.

#### 2.5. System Design

In this paper, the use of the KNN algorithm to predict the potential for heart disease required an interface as an input and output medium. Website technology is the ideal interface, as everyone can access it without the need to having an executable and can be accessed anywhere and on any platform. In developing this website technology, the Flask web framework was used and written in Python. Flask provides convenience and flexibility in development and it is lightweight because there are few built-in libraries, so it is called a micro-framework [27].

The system was designed using the waterfall method because the system built has clearer specifications and steps. Thus, the system will minimize errors in both the development and usage of this method, with consistent focus on each development process. According to Gumawang and Rakhmadi [28], this method has several stages, including: needs analysis, design, development, testing, implementation, and maintenance. These stages are described in the flow chart in Figure-2.



**Figure 2-** Waterfall model implemented in the system

**A. Requirements Analysis**

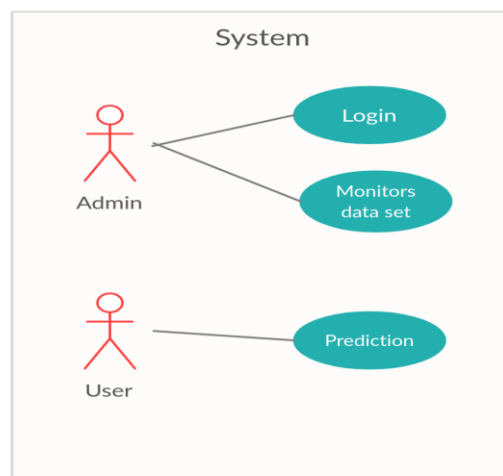
In developing this research, it did not require a lot of needs in terms of information systems because the data needed is a data set that has been collected UCI Machine Learning Repository. The tools used were the Flask micro-framework which uses the Python programming language because it is considered light and does not consume a lot of resources, so that resources can be diverted for the prediction process by the KNN algorithm. Other data required is the predictable user data that will be obtained when the system is on the air.

**B. Design**

The system was designed using two Unified Modelling Language (UML) models to direct the system during the development stage, which are the Use Case and Activity Diagram models. Use Case Model is a representation of the interaction between users and the system. Activity Diagram model is a workflow representation of system activities.

**I. Use Case**

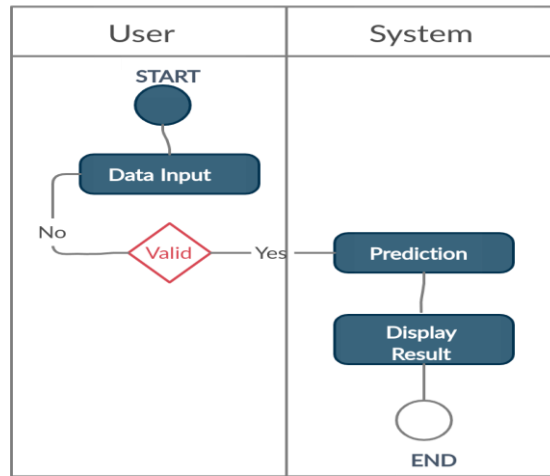
Use Case is a diagram depicting the role of an actor in certain functions [29]. The use of this diagram is to facilitate system development. In this system, there will be 2 actors, user and admin. Users are ordinary users who can perform tests to determine the potential for heart disease by entering the data required by the system. Admin is the manager who monitors the dataset.



**Figure 3-** Use case diagram of the system

**II. Activity Diagram**

Activity Diagram is a flow chart of the process of using the system by actors and how the system responds to the actors' actions.



**Figure 4-** Activity diagram of the system

The Activity Diagram in Figure- 4 shows that users need to enter the data needed by the system to perform predictive analysis. If the data entered is incorrect or not as requested, an error message will appear and ask the user to enter the data correctly. If the data is correct, the system will make predictions, display the results, and store them in the database.

**2.6. System Development**

The system that has been designed will be developed as a web-based application. The application will be developed using several tools to improve its efficiency. The implementation of web framework is expected to prevent security issues and to reduce inefficient code, i.e. the repeated writing of the same function. The tools used in this paper are explained as follows.

**A. Python**

The system was developed using the Python programming language. Python is one of the popular programming languages used for web application development. According to the Tiobe Index, Python ranks 4th as the most popular programming language as per June 2016 [27].

**B. Flask**

Flask is a micro-framework written in the Python programming language which does not have many tools and libraries [15]. Flask is used to make development more efficient, . Using Flask will ease the burden on resources and can be diverted for KNN.

**C. MySQL**

MySQL is a Database Management System (DBMS) that can be used for data storage on the system. MySQL is used in many systems due to its speed in processing data. It has a Structured Query Language (SQL) that is easy to understand so that it supports the efficiency of system development [30].

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
267	49	1	2	118	149	0	0	126	0	0.8	2	3	2
178	43	1	0	120	177	0	0	120	1	2.5	1	0	3
289	55	0	0	128	205	0	2	130	1	2.0	1	1	3
101	59	1	3	178	270	0	0	145	0	4.2	0	0	3
261	52	1	0	112	230	0	1	160	0	0.0	2	1	2

**Figure 5-** Data before Z-score normalization

**2.7. System Testing**

The low complexity of this system does not require many and complicated tests on the system. The use of black box testing is sufficient to test the features and workflows of the system, considering that the system does not have many features. Black box testing performs pragmatic testing by looking at the results of each action the tester applies to the system. Observing the response given by the system is as expected or not. The functions tested are those that deal directly with the users, such as login, prediction, or data input. It can be performed manually or using the unit testing method.

**3. Results and Discussion**

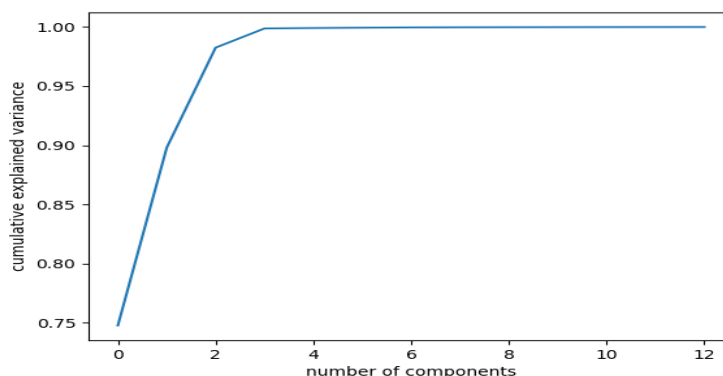
A total of 303 data collected from the dataset that has 13 dimensions and 1 label. A total of 54.45% or 165 patients had heart disease and 45.54% or 138 patients did not suffer from heart disease. Before implementing the data on the website, it had to be processed using the KNN algorithm to find the most optimal accuracy results.

The initial step was to divide the data set into training data and testing data with a composition of 70% for training data and 30% for testing data. The amount of data for training data was 212 data and the amount of data for testing data was 91 data. In the testing data, there were 50 data of heart disease patients and 41 data of non-heart disease patients. The training data was carried out through the first stage of data preprocessing, namely normalizing the dataset using Z-score Normalization. Since the data scale is different, it might cause inequality in the weight of each dimension. Z-score Normalization will be useful in increasing accuracy [18]. Data with a value below the average will have a negative value, while data with a value above the average will have a positive value. As shown in Figure- 5, data that was not normalized appeared to have a different scale in each dimension. Figure- 6 shows that the data that was normalized had a more balanced scale in each dimension.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
207	-0.593023	0.635589	0.971638	-0.788130	-1.930925	-0.429369	-0.974965	-1.092298	-0.724693	-0.204956	1.011443	2.319723	-0.599833
208	-1.278150	0.635589	-0.944522	-0.676629	-1.387987	-0.429369	-0.974965	-1.360057	1.379895	1.290311	-0.600782	-0.678943	1.142149
209	0.092104	-1.573344	-0.944522	-0.230627	-0.845048	-0.429369	2.817558	-0.913792	1.379895	0.850526	-0.600782	0.320612	1.142149
210	0.548856	0.635589	1.929718	2.556886	0.415344	-0.429369	-0.974965	-0.244394	-0.724693	2.785578	-2.213008	-0.678943	1.142149
211	-0.250459	0.635589	-0.944522	-1.122632	-0.360282	-0.429369	0.921297	0.425005	-0.724693	-0.908611	1.011443	0.320612	-0.599833

**Figure 6-** Data after Z-score normalization

The previously normalized data were preprocessed, then the data dimension was reduced using PCA, because of the large number of dimensions (13 dimensions). When carrying out the reduction process with PCA, it is necessary to determine how many principal components that are generated. The principal component value chosen is the one that does not lose much information (information loses). In Figure- 7, it appears that PC 2 had a cumulative variance of 98.15% and reduced 84.61% of the total dimensions, or as many as 11 dimensions. The PC 2 value was chosen because, with the cumulative variance of 98.15%, there was not much information loss from the original data.



**Figure 7-** Value of Principal Component

The next stage was data processing using the KNN algorithm for classification. The value of K as a neighbor must be an odd number because the numbers of classes were even. To find the optimal value of K, k-fold cross validation was utilized with k on the k-fold was 10. The value of K for the KNN used a range of  $1 \leq K \leq 20$ . In Figure- 8, the value of K=7 had the highest level of accuracy compared to other values of K.

The value of K = 7 is obtained if the amount of data in the dataset is 303 or the maximum is added by 19 data to 322. If the additional data exceeds 19 and the total dataset exceeds 322, then the K value is no longer 7.

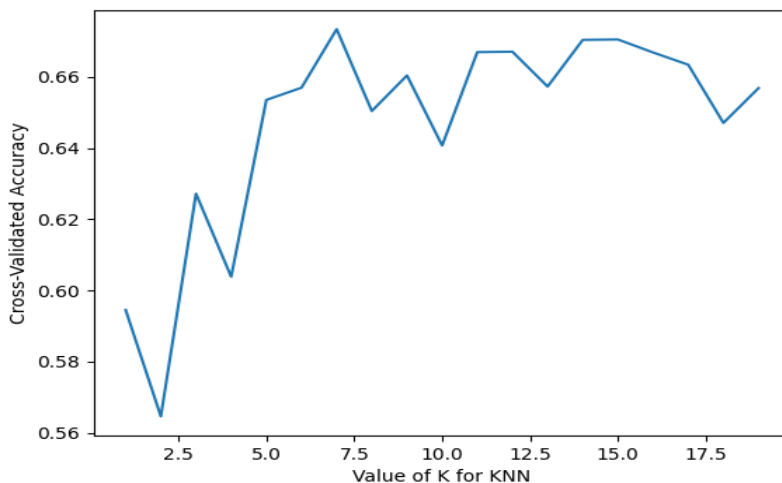


Figure 8- Value of K for KNN

Unlabelled testing data that was achieved by the prediction process was compared with the target testing data and the average match of the two data was found to compute its accuracy. The accuracy resulted from the KNN process after normalization is higher than that in the KNN process without normalization [31]. In this paper, we show that the accuracy generated by the KNN algorithm with normalized data using the Z-score normalization had a higher value (82.41%), as explained in table 2.

Table 2-KNN accuracy

Data	Accuracy
Normalization	82.41%
Without Normalization	65.93%

As shown in Figure- 9, the website system was designed as an interface to predict the potential for heart disease in the implementation of the KNN algorithm. The system works in one direction, that is; it accepts input from users and displays prediction results that have been processed using the KNN algorithm based on data entered by users through the website. Flask is a micro-framework, so it does not have many built-in libraries. To support the efficiency of website system design, several modules were used, including WT Forms, SQLAlchemy, Bcrypt, and Login.

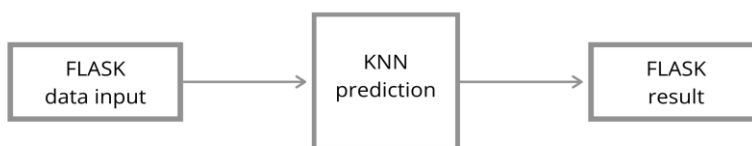
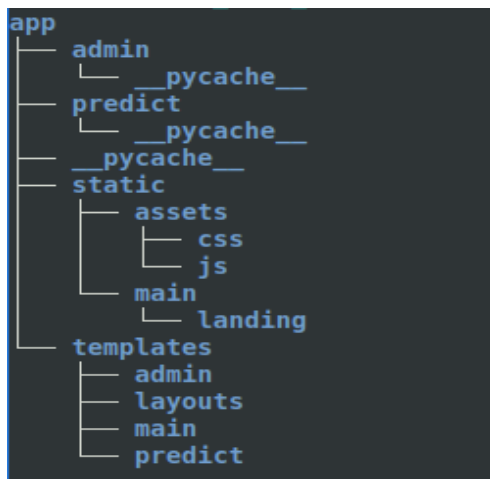


Figure 9- System workflow



Flask allows the development of websites with a more flexible file structure. In Figure- 10, the structure of the website application was located in the app directory where the admin and predict directories are used to place various business logics for admin and prediction process. Static directory contains assets for website display in the form of CSS and Javascript code. The html code that displays various data is located in the templates directory using the Jinja2 template engine so as to provide efficiency and tidiness in the code written and more orderly data delivery from the view to the template.



**Figure 10-** Directory structure of the system

The website will receive input on route /predict and be processed on the same route. By using the KNN function that was prepared, the user input data will be predicted using the KNN algorithm with the dataset in the database using the MySQL DBMS.

Any data that was predicted and its results will be entered into the database so that it will multiply the existing dataset in the database. The new data in the database is gathered into a dataset and the training process will be carried out for every new data that comes in. For each periodic learning process, the quality of KNN model continues to improve. It is called dynamic learning, in which the model is machine learning that is dynamically updated and periodically improved according to the new data obtained.

After the prediction process is complete, the system will redirect to route /predict/result and display a page that provides information on the prediction results and recommendations for a healthy lifestyle. Users can access route /heart to find out information related to heart disease and its prevention.

The whole website system has nine routes, as shown in table 3. There is one admin who will monitor how many users have used the website system. As shown in table 4, the database has two entities, namely data set and admin. The data set entity is in the form of a table containing the dataset that will be used by KNN, while the admin entity is in the form of a table containing admin data for authentication.

**Table 3-**Website routes of the system

Method	Route	Information
GET	/	Home page
GET & POST	/login	Login admin
GET	/logout	Logout admin
GET	/home	Admin page
GET & POST	/predict	Data input by users and the prediction process
GET	/predict/result	Prediction result display

GET	/guide	Guide page of the system
GET	/heart	Heart-related information
GET	/about	System-related information

**Table 4-** Database of the system

Table	Attribute	Information
Admin	id, username, email, password	Data admin
Dataset	id, age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, target	Data set

The routes /login, /logout, and /home handle processes that occur on the admin side. The route /login displays the login page and the authentication process, while /logout has a method to delete the session when the admin leaves the admin page, and /home is the admin home page that displays data of users who have successfully made predictions.

The routes /predict and /predict/result display the user data input page and display the prediction results. The routes /about, /heart, and /guide contain information about the system, heart disease, and instructions for using the system.

The last stage is testing the system using the black box testing method, as shown in table 5. In this test, the system will be tested by exposing it to active users and it must be able to respond as planned.

**Table 5-**Black box testing result

Function	Input	Output	Status
Home page	Access website	Display home page	Valid
Login page	Input correct email and password	Display admin page and data of user	Valid
	Input incorrect email and password	Back to login page and display error message	Valid
Prediction	Input data of user	Display results of the prediction	Valid
Information about heart disease and system	Access route /about, /guide, and /heart	Display information about heart disease, system, and system use	Valid

#### 4. Conclusions

From the paper, it is found that the KNN algorithm could be used well in implementing web-based systems using Flask Python. The use of Z-score normalization greatly affected the prediction results until reaching an accuracy of 82.41%. The implementation of the algorithm into a website system also eased the use of the prediction process of potential heart disease and could be used by any user.

The prediction system built on a website can be accessed through various media and users only need to visit the address of the system without installing it on the user's device.

To support more optimal efficiency and effectiveness of the prediction system in further research, it is necessary to increase the learning process by modifying algorithms or conducting a better preprocessing data, as well as improvising the data set because it will be better if the data set has more data with less noise. The system will be more efficient if the machine learning algorithm can optimize every new data obtained and improvise the algorithm independently. In further research, the improvement considered is that the system is not only able to predict the risk of heart disease but also to predict heart disease cases directly, with detailed information on what types of heart disease will be predicted.

## References

1. Widyastuti, D. A. **2018**, December 17. Pengaruh Kebiasaan Konsumsi Junk Food Terhadap Kejadian Obesitas Remaja [The Effect of Junk Food Consumption Habits on Adolescent Obesity]. <https://doi.org/10.31219/osf.io/7d8ey>
2. Sutrisno, S., Pratiwi, D. C., Istiqomah, I., Baba, K. J., Rifani, L. E., & Ningtyas, M. A. **2018**. Edukasi Bahaya Junk Food (Makanan dan Snack) dan Jajan Sembarangan dikalangan Remaja [Education on the Dangers of Junk Food (Food and Snacks) and Casual Snacks among Teens]. *Journal of Community Engagement in Health*, **1**(1): 7-10.
3. Ministry of Health of the Republic of Indonesia, **2014**. Pusat Data Dan Informasi Kementerian Kesehatan Ri. In: Situasi Kesehatan Jantung [Data And Information Center Of The Ministry Of Health Ri. In: Heart Health Situation]. Jakarta: Kementerian Kesehatan RI, pp. 2.
4. Rohman, A., Suhartono, V., & Supriyanto, C. **2017**. Penerapan Algoritma C4.5 Berbasis Adaboost Untuk Prediksi Penyakit Jantung [Adaboost-based C4.5 Algorithm for Prediction of Heart Disease]. *Cyberku Journal*, **13**(1): 2-2.
5. Iskandar, I., Hadi, A., & Alfridsyah, A. **2017**. Faktor Risiko Terjadinya Penyakit Jantung Koroner pada Pasien Rumah Sakit Umum Meuraxa Banda Aceh [Risk Factors for Coronary Heart Disease in Patients at Meuraxa General Hospital Banda Aceh]. *AcTion: Aceh Nutrition Journal*, **2**(1): 32-42.
6. Ghani, L., Susilawati, M. D., & Novriani, H. **2016**. Dominant Risk Factors Of Coronary Heart Disease In Indonesia. *Buletin Penelitian Kesehatan*, **44**(3): 153-164.
7. Sukaman. **2019**, October 3. Mengenal Gangguan Irama Jantung (Aritmia) Serta Bagaimana Pencegahan dan Penanganannya [Get to know heart rhythm disorders (arrhythmias) and how to prevent and handle them]. Retrieved from <https://pjhk.go.id/artikel/mengenal-gangguan-irama-jantung-aritmia-serta-bagaimana-pencegahan-dan-penanganannya>.
8. Yuniadi, Y. **2017**. Mengatasi Aritmia, Mencegah Kematian Mendadak [Overcoming Arrhythmias, Preventing Sudden Death]. *eJournal Kedokteran Indonesia*, **5**(3): 46-139.
9. Burmaoglu, S., Sartenaer, O., & Porter, A. **2019**. Conceptual definition of technology emergence: A long journey from philosophy of science to science policy. *Technology in Society*.
10. Rayan, Z., Alfonse, M., & Salem, A. B. M. **2019**. Machine Learning Approaches in Smart Health. *Procedia Computer Science*, **154**: 361-368.
11. Budianto, A., Ariyuana, R., & Maryono, D. **2018**. Perbandingan K-Nearest Neighbor (Knn) Dan Support Vector Machine (Svm) Dalam Pengenalan Karakter Plat Kendaraan Bermotor [Comparison Of K-Nearest Neighbor (Knn) And Support Vector Machine (Svm) In Introduction To Plat Vehicle Characters]. *Jurnal Ilmiah Pendidikan Teknik dan Kejuruan*, **11**(1): 27-35.
12. Anggoro, D. A., Rahmatullah, P. I. **2020**. The Implementation of Subspace Outlier Detection in K-Nearest Neighbors to Improve Accuracy in Bank Marketing Data. *International Journal of Emerging Trends in Engineering Research*, **8**(2).
13. Mutrofin, S., Izzah, A., Kurniawardhani, A., & Masrur, M. **2015**. Optimasi teknik klasifikasi modified k nearest neighbor menggunakan algoritma genetika [Optimization of the modified k nearest neighbor classification technique uses a genetic algorithm]. *Jurnal Gamma*, **10**(1).
14. Andiani, L., Sukemi, S., & Rini, D. P. **2020**, February. Analisis Penyakit Jantung Menggunakan Metode KNN Dan Random Forest [Analysis of Heart Disease Using the KNN and Random Forest Method]. In **Annual Research Seminar (ARS)**. **5**(1): 165-169.
15. Chauhan, N., Singh, M., Verma, A., Parasher, A., & Budhiraja, G. **2019**. Implementation of database using python flask framework. *International Journal of Engineering and Computer Science*, **8**(12): 24894-24899.
16. Aini, S. H. A., Sari, Y. A., & Arwan, A. **2018**. Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes [Selection of Information Gain Features for Classification of Heart Disease Using a Combination of the K-Nearest Neighbor and Naïve Bayes Methods]. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN, 2548, 964X.
17. Gunawan, D. **2016**. Evaluasi Performa Pemecahan Database dengan Metode Klasifikasi Pada Data Preprocessing Data mining [Evaluation of Database Splitting Performance with Classification

- Methods in Data Preprocessing Data mining]. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, **2**(1): 10-13.
18. Anggoro, D. A., & Supriyanti, W. **2019**. Improving accuracy by applying Z-score normalization in linear regression and polynomial regression model for real estate data. *International Journal of Emerging Trends in Engineering Research*, **7**(11): 549-555.
  19. Kusnadi, A., & Ranny, R. **2016**. Identifikasi Dini Kerusakan Jalan Flexible Pavement Dengan Menggunakan Algoritma PCA [Early Identification of Damage to Flexible Pavement Roads Using the PCA Algorithm]. *Ultimatics: Jurnal Teknik Informatika*, **8**(2): 125-130.
  20. Afrin, F., & AT, M. **2015**. Comparative Performance Of Using PCA With K-Means And Fuzzy C Means Clustering For Costomer Segmentation. *International Journal Of Scientific & Technology Research*, **4**: 70-74.
  21. Jamal, A., Handayani, A., Septiandri, A. A., Ripmiatin, E., & Effendi, Y. **2018**. Dimensionality Reduction using PCA and K-Means Clustering for Breast Cancer Prediction. *LONTAR KOMPUTER: Jurnal Ilmiah Teknologi Informasi*, 192-201.
  22. Adiwijaya, W. U., Lisnawati, E., Aditsania, A., & Kusumo, D. S. **2018**. Dimensionality Reduction using Principal Component Analysis for Cancer Detection based on Microarray Data Classification. *Journal of Computer Science*, **14**(10).
  23. Nasution, M. Z. **2019**. Penerapan Principal Component Analysis (Pca) Dalam Penentuan Faktor Dominan Yang Mempengaruhi Prestasi Belajar Siswa (Studi Kasus: Smk Raksana 2 Medan) [Implementation Of Principal Component Analysis (Pca) In Determining Dominant Factors Affecting Student Learning Achievement (Case Study: Raksana 2 Medan Vocational High School)]. *JurTI (Jurnal Teknologi Informasi)*, **3**(1): 41-48.
  24. Amra, I. A. A., & Maghari, A. Y. **2017**, May. Students performance prediction using KNN and Naïve Bayesian. In 2017 8th International Conference on Information Technology (ICIT) (pp. 909-913). IEEE.
  25. Nilashi, M., Ibrahim, O., Dalvi, M., Ahmadi, H., & Shahmoradi, L. **2017**. Accuracy improvement for diabetes disease classification: a case on a public medical dataset. *Fuzzy Information and Engineering*, **9**(3): 345-357.
  26. Lutfi, M., & Hasyim, M. **2019**. Penanganan Data Missing Value Pada Kualitas Produksi Jagung Dengan Menggunakan Metode K-Nn Imputation Pada Algoritma C4.5 [Handling Of Missing Value Data On Corn Production Quality Using K-Nn Imputation Method On C4.5 Algorithm]. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, **2**(2): 89-104.
  27. Vogel, P., Klooster, T., Andrikopoulos, V., & Lungu, M. **2017**, September. A low-effort analytics platform for visualizing evolving Flask-based Python web services. In 2017 IEEE Working Conference on Software Visualization (VISSOFT) (pp. 109-113). IEEE.
  28. Gumawang, A. K. A., & Rakhmadi, A. **2018**. Pengembangan Sistem Perancangan Manajemen Usaha Kecil Menengah Bidang Kuliner dengan Metode Swot [Development of Management Design Systems for Small and Medium Culinary Enterprises using the Swot Method]. *Proceeding of The URECOL*, 159-170.
  29. Shofia, S., & Anggoro, D. A. **2020**. Sistem Informasi Manajemen Administrasi Dan Keuangan Pada Tk-It Permata Hati Sumberrejo-Bojonegoro [Administration And Financial Management Information System At Tk-It Permata Hati Sumberrejo-Bojonegoro]. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, **5**(2): 221-230.
  30. Anggoro, D. A., & Supriyanti, W. **2019**. Aplikasi Sistem Pendukung Keputusan dengan Metode AHP untuk Pemilihan Siswa Berprestasi di SMAN Kebakkramat [Decision Support System Application with AHP Method for Selection of Outstanding Students in Kebakkramat High School]. *Jurnal Penelitian dan Pengabdian Kepada Masyarakat UNSIQ*, **6**(3): 163-171.
  31. Anggoro, D. A., & Kurnia, N. D. **2020**. Comparison of Accuracy Level of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) Algorithms in Predicting Heart Disease. *International Journal*, **8**(5).