# Chapter 4
# Design Analysis

## Analysis

The topic that I use for this project is related to the Internet of Things. First I took data using the DHT11 sensor and Arduino UNO. The data was obtained using the Arduino default application, namely the Arduino IDE (Integrated Development Environment). How to retrieve data using an IP Address and a LAN cable that is already connected between the Router and Arduino UNO. The data that has been obtained is divided into 3 classes, namely temperature, humidity, and rhythm, with the target class, namely healthy, which is connected to the PHP code to be entered into the XAMPP localhost which functions as a database of the data that has been obtained. Then the data that is in the localhost XAMPP is sent to PHP Code in CSV format for processing. After the data is sent into the PHP Code, data filtering is carried out to obtain optimal data. The data filtering is carried out using the Naive Bayes Classifier method, namely by looking for the probability of the data that has been split. I use 500 data which is divided into Training Data and Testing Data to predict the other 500 data. The training and Testing that I use a ratio of 80%: 20%. The first 500 data are divided into 450 Training Data and 50 Testing Data. After the final results appear in the form of Accuracy, Precision, Recall, and F1-Score, the next step is to make predictions for the next 50 data to determine whether the healthy patient is YES or NO. The 50 existing data already have actual data, so the data prediction function performed is to find out the Confusion Matrix. After that, the next 500 blank data are predicted to determine whether the obtained target class is YES or NO.

## Design

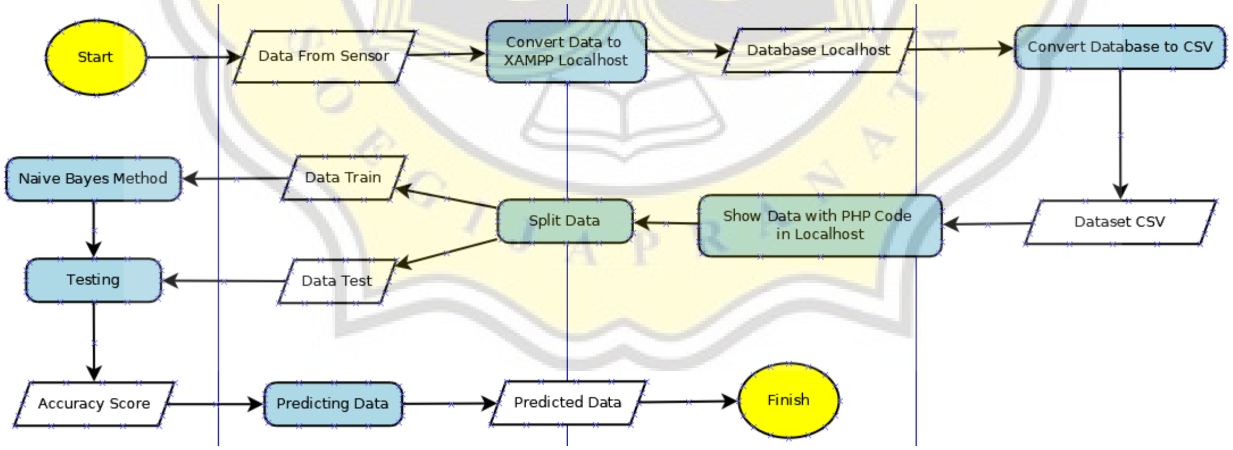

Figure 6: Sensor System Design

### 2.1 Naive Bayes Scenario

In the design of this tool, I use the Naive Bayes algorithm to find the probability of the existing dataset. Initially, the existing dataset is divided into 2 parts, namely the dataset that already contains the target and the dataset that does not yet contain the target. After dividing by two, we divide the dataset that has been filled with targets into 2 parts, namely train data and test data. After being divided into 2 parts, we predict the test data that is tested based on the existing train data. When the prediction data has appeared, then we compare it with the actual data available, to produce a Confusion Matrix which will be used to obtain the Accuracy Score. After the dataset that already contains the target gets the Accuracy Score, then we continue to predict the dataset that does not have a target. The way to predict it is to find the probability of the target train used, if the results appear, that's the prediction data. After the prediction data appears, the data is separated into 2 parts, namely the train data and the existing test data. Later, if we have gone through the testing process and get the results of the existing Accuracy Score, we will compare the Accuracy Score from the first dataset and the second dataset. This is done to determine whether the accuracy score obtained has obtained optimal results, overfit, or underfit. The accuracy score of the second dataset obtained is also used to determine whether the data obtained is better or worse.

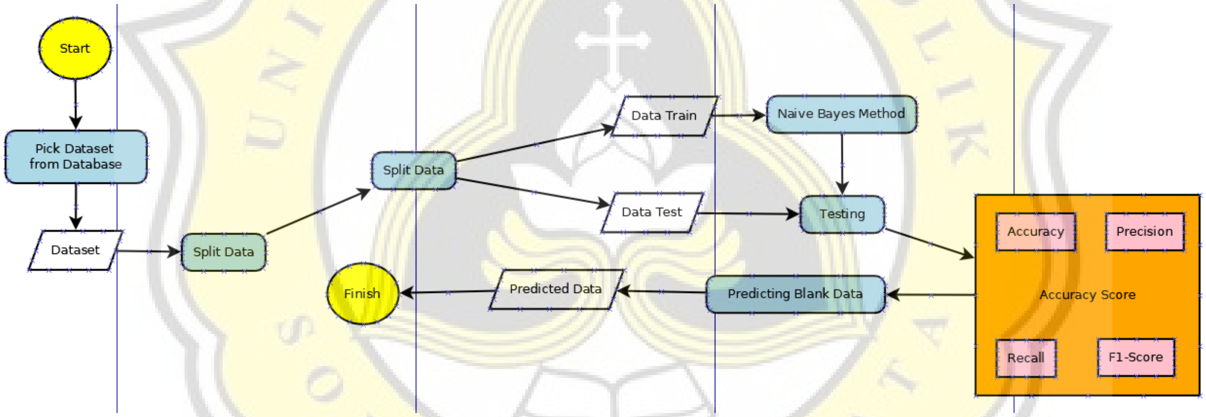

Figure 7: Naive Bayes Scenario