

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

The problem of this research is because many diseases are sharing the same disease so that it is hard to make a prediction using inputted symptoms using if or switch case in the code. The dataset that collected using web scraping has 1848 rows. After data stored only the unique one, it drops to 1592 rows or symptoms. That's why we know that disease shares the data with another disease.

Table 1: Disease that has these keywords

Keywords	Total Disease
Batuk	36
Bengkak	62
Demam	62
Mual	39
Nyeri	235
Penurunan berat badan	18
Sesak Napas	23
Sakit Kepala	26

The above data is showing what kind of data we analyzed. There is 2 column, keywords are a string that we search in every symptom inside disease and the other is Total Disease, that is how many diseases that have the left keyword. In those 8 samples, we know that every keyword appears in another disease, the highest one is "Nyeri" that includes 235 diseases.

To handle the data and make a prediction system, an algorithm is required. This research is using the Naive Bayes Algorithm to calculate the data. Naive Bayes is a classifier that is predicting future results based on past data.

Above is a Naive Bayes formula. Here is the detail, $P(A_i | A_j)$ is a probability A_i if know A_j condition. N_c store 1 or 0, where 1 if symptoms available in disease, and 0 if symptoms not available in disease. M is total symptoms and “ p ” is a disease total.

Naive Bayes calculation is divided into 3 parts, first is determining whether N_c is 1 or 0, then calculating input user based on N_c score, and the result should be under 1, the last is times all the second steps and times again with 1 divided by disease total.

4.2 Desain

Here is the Flow Chart of this system:

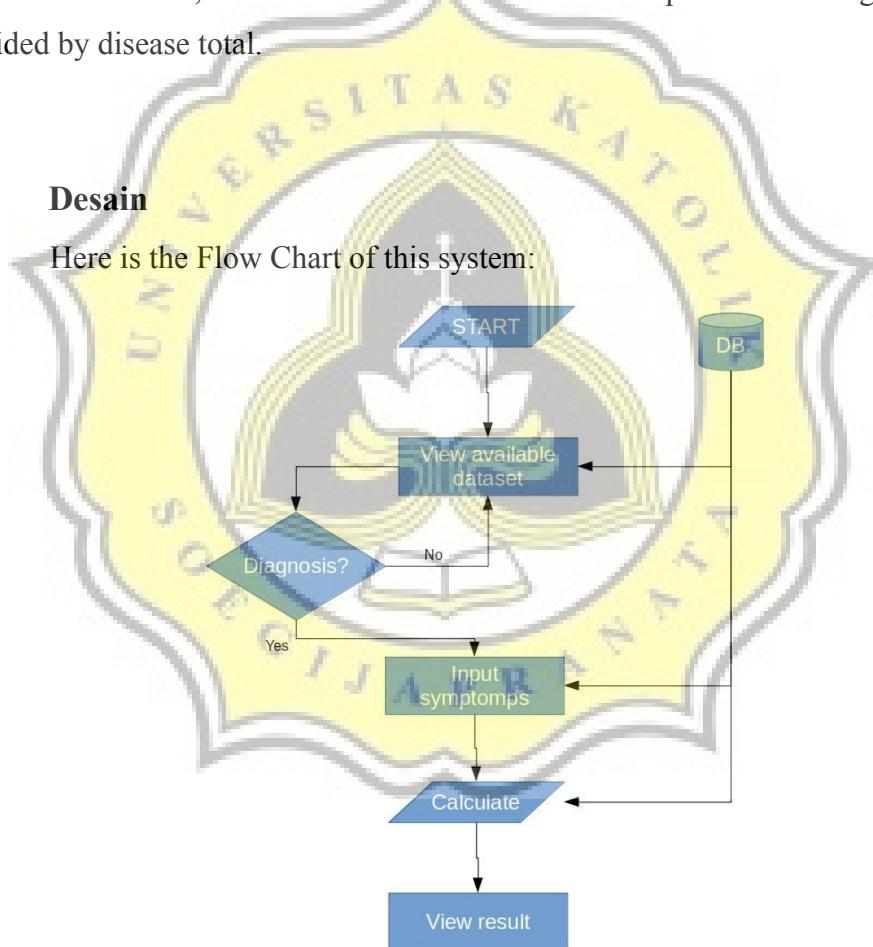


Illustration 2: Flow chart

The user starts opening the website. The first webpage that is seen is View available dataset. In that webpage, users can see every disease and symptoms that are supported by the system. The next view is whether the user wants to go to the

input symptoms and redirected to the symptoms list, or stay on the home page views any of our data. This page has required a user to pick come off the checkbox and if done, the user can click the submit button to see the result. Finally, the result page will show the user what is the recommendation data based on the algorithm.

The database uses SQLite as it is the default database from Django. From the problem above, the database is having 3 tables. Table disease has the disease name and its id. Next, are symptoms that have name and id_area. The last is a join between disease and symptoms.

