CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

Ordinary people need calories every day that must be met because calories are source of energy for everyone. The amount of calories obtained from the amount of nutrient in the food comsumed. For diabetics consumption of food that is needed is different from the general public, diabetics must consume foods that contain nutrients with certain limits but still meet their calorie needs. Well for people, especially diabetics, it is usually difficult to determine what to consume, such as what staple foods are then with what kind of side dishes and vegetables and what kind of fruit is recomended, so that the amount of nutrients from food consumed can meet the required calories needs, but stick to existing restrictions. Therefore, thes existence of this project is useful to help in determining the compositions of food with diffeent combinations, but still pay attention to the recomended restrictions.

The amount of calories needed for diabetics for men is different with the amount of calories for women. Following are the calculations to determine the amount of calories for men and women.

Calculation calories formula for men :

$$66+(13{,}7*berat\,badan)+(5*tinggi\,badan)-(6.8*usia)$$

Calculation calories formula for women :

$$655+(9.6*berat\,badan)+(1{,}7*tinggi\,badan)-(4{,}7*usia)$$

Then after we got the amount of calories needed for diabetics, so the next is to find out the limit of the nutritional content of each food consumed. The following are ways to determine the limits of each nutrient in a food for diabetics patient.

Table 4.1: Calculation Nutrition Formula Table

Name of Nutrition	**Minimum**	**Maximum**
Carbohydrates	$\frac{0{,}45 * total\ calories}{4}$	$\frac{0{,}65 * total\ calories}{4}$
Fat	$\frac{0{,}2 * total\ calories}{9}$	$\frac{0{,}25 * total\ calories}{9}$
Proteins	$\frac{0{,}1 * total\ calories}{4}$	$\frac{0{,}15 * total\ calories}{4}$

In addition to nutrition in the table 4.1 above there are still other nutrients contained in food, where the way to determine the limits are different. Here are the nutrients and their limitations.

Table 4.2: Other Nutritional Limits

Name of Nutrition	**Requirements**
Natrium	$2 \geqslant total\ nutrition \leqslant 2.3$
Fiber	$20 \geqslant total\ nutrition \leqslant 35$
Glucose	$20 \geqslant total\ nutrition \leqslant 50$

After explaining the calculation of calories and nutritional limits, it will now discuss the data storage section. In this project, storage uses arraylist data structures. The way of storage is done by grouping foods according to the categories of food. In this project the categories are divided into 4 types, namely staple foods, side dishes, vegetable, and fruit. Examples of basic food categories are only for storing staple foods, side dishes are only for storing side dishes, etc. There are 51 food data in this project. The data used using units per portion. The following are examples of food data that used in this project.

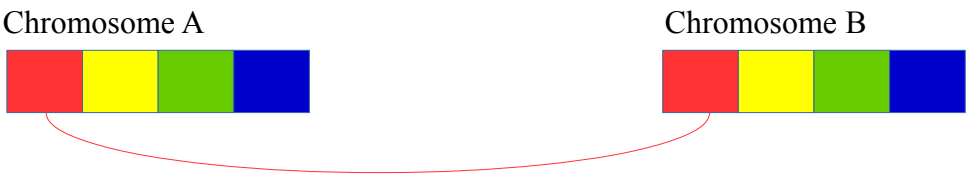
Table 4.3: Sample of Food Data

Food Name (portion)	Oatmeal	Telur Dadar	Sayur Asem	Pisang
Carbohydrates (g)	17.02	0.345	12.9	28.78
Fat (g)	1.6	3.005	2.76	0.42
Protein (g)	4.07	5.31	3.18	1.37
Natrium (g)	0.187	0.0805	0.308	0.001
Fiber (g)	12.5	0.0	2.5	3.3
Glucose (g)	0.38	0.325	4.75	15.41

Calorie(kcal)	97.0	76.5	80.0	112.0

After storing data into arraylist, it now goes to the part of the genetic algorithm. First is generating random populations, the population model used is 1 population consisting of 4 chromosomes. Each chromosome represents each food category. Then goes to the evaluation section. Evaluation section is a calculation of the value of fitness. The fitness value here serves to show the chromosome quality of a population. The formula used in this project to calculate the fitness value is : $\frac{1}{\sum(total\ nutrition)}$. After getting the best value of fitness, crossover is done to get a new chromosome by crossing of 2 randomly chromosomes with a cut point. The crossover method used *One-Point Crossover*. The way is make a new chromoshome with the initial contents is the parent till the cut point and the rest is filled with child.

The result of crossover process is used for the next process, namely the mutation. Mutations are carried out by randomly exchanging individuals on a same chromosome. The mutation method used *Exchange Mutation*. But the mutation in this project is different because the storage carried out in only 1 population of 4 chromosomes with each category, then the mutation are carried out by exchanging individuals with 2 chromosomes randomly with the condition that individual exchanges must be in the same food category.



So the result of crossover and mutation will form a new chromosome. Then after the crossover and mutation process then go to the selection stage. The chromosomes used in this selection stage are the new chromosomes which are the result of crossover and mutation processes. There are 2 types of selection used in this project, namely Elitism Selection and Roulette Wheel Selection, so later the user can choose by themselves selection wants. Elitism Selection is selection that looks for the best individuals and maintains them, so for the next generation this individual will be remain. Whereas the Roulette Wheel Selection have an idea that each individual has the same opportunity to be an individual for the next population. If the value of fitness greater than the value of random number, so the individual will goes to the next generation as the initial population. This evaluation (calculation value of fitness), crossover, mutation, and selection processes are repeated until reaches the maximum limit of the specified generation.

4.2 Design

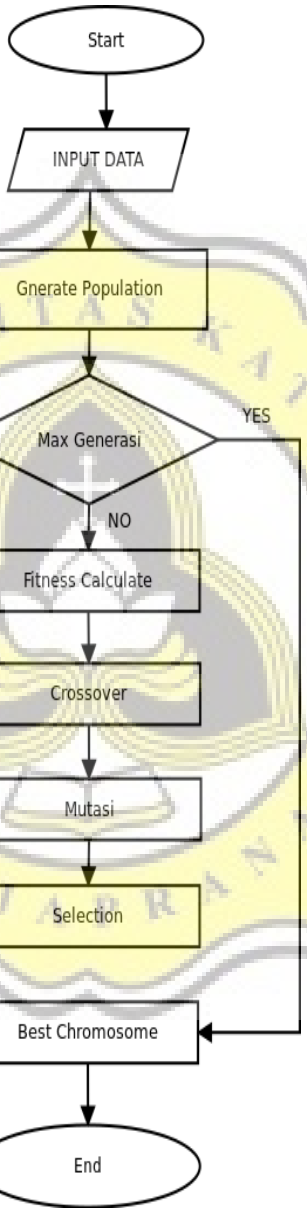


Illustration 4.1: Flowchart Determining Food for Diabetics

In illustration 4.1 above shows the flow of how the program works from this project. Firstly, fill in the data needed in the form of name, gender, age, weight,

height, many generations, and the desired selection method. Then the program will continue to generate random populations. After that, the best value of fitness is calculated. Then will proceed to the crossover process by generating 2 chromosomes randomly and do the the crossing. The crossover is done, then goes to mutation process. The mutation process is done by generating 2 random chromosomes and exchange the individual that have same type. The last steps is to make a selection process to choose the best population for the next generation. All of those processes are carried out as many generations number.

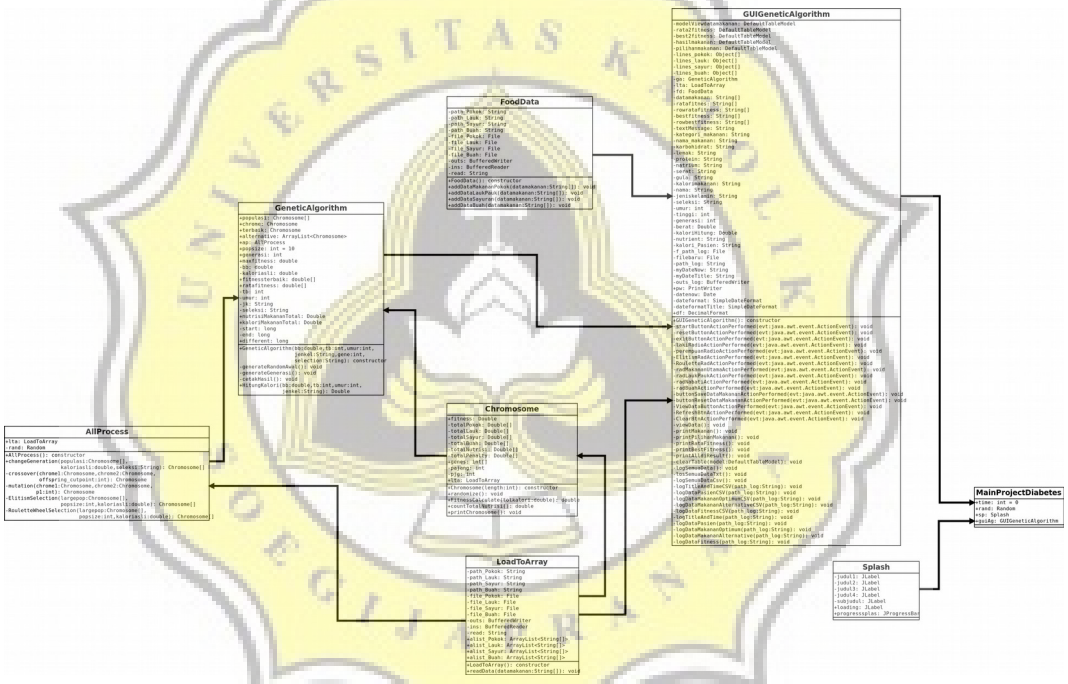


Illustration 4.2: Class Diagram

From the class diagram in illustration 4.2 it can be seen that in this project there are 8 classes including main class. The GuiGeneticAlgorithm uses the FoodData class, the LoadToArray class, and the GeneticAlgorithm class. The FoodData class serves to store new food data that is entered by user. The LoadToArray has the function to read data and then load it to arraylist. The GeneticAlgorithm class has the function to carry out the overall genetic algorithm function. The GeneticAlgorithm class itself uses the Chromosome class and the AllProcess class. The Chromosome class has a function for making chromosomes,

intial populations, calculated value of fitness, and print data. The AllProcess class has the function to perform all the processes of the genetic algorithm, namely crossover, mutation, slection and change generation. Last is the LoadToArray class instead of being used in the GeneticAlgorithm class, it also used in the Chromosome and AllProcess class.