

## CHAPTER IV DESIGN AND ANALYSIS

### 4.1. Analysis

The goal of this project is to generate a crossword grid based on user's words. So the program requires sets of words and its clue as its input. Besides set of words and its clue, another input required such as number of population, chance of mutation, and number of generation. This data will become user's input, so that the user can experiment with the algorithm freely. All of the required data will be stored in an object. This object will be used to make parsing data from classes easily. In this project a chromosome will become an object that consist of object Gen, it will be discussed later in initialization and encoding process.

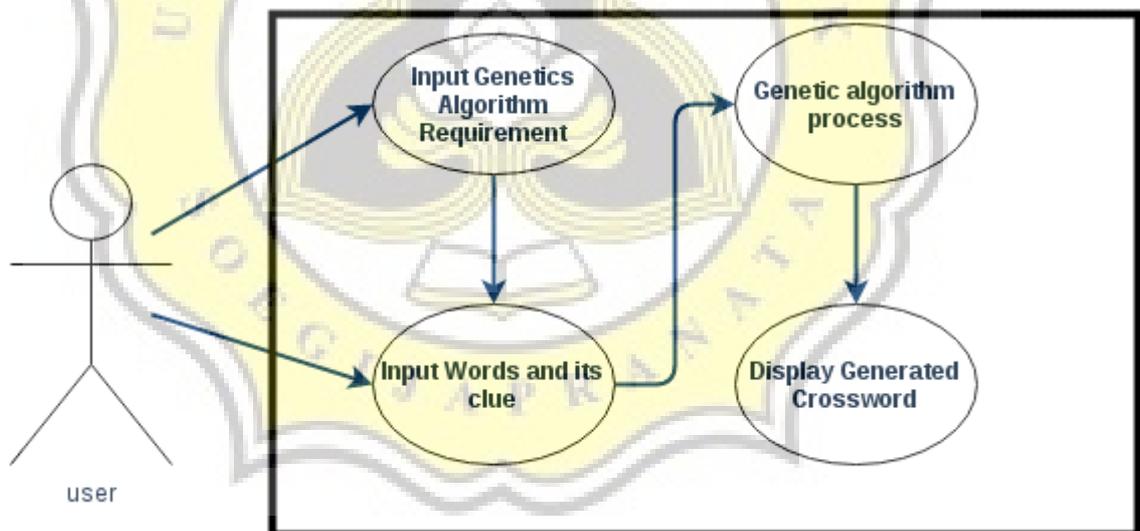


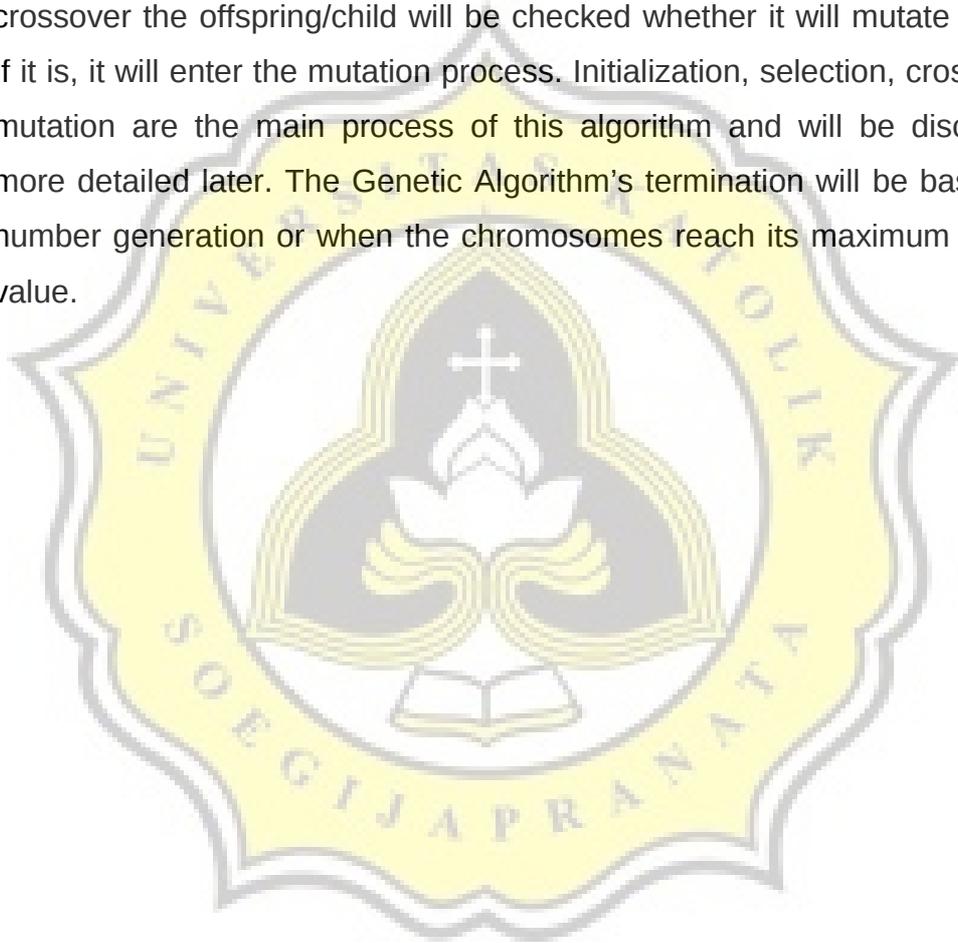
Figure 2. Use Case Diagram

### 4.2. Design

After analyze the problem, a design needed to implement Genetic Algorithm. There are some processes in Genetic Algorithm. Design of each process will be explained later in sub chapter.

#### 4.2.1. Genetic Algorithm's Process

The Genetic Algorithm used in this project is steady-state Genetic Algorithm. This process will keep the variation of chromosome in a population. After the input stored, the program will start the initialization process to make the first population. After initialization, the parent for the crossover process will be selected in selection process. After the crossover the offspring/child will be checked whether it will mutate or not. If it is, it will enter the mutation process. Initialization, selection, crossover, mutation are the main process of this algorithm and will be discussed more detailed later. The Genetic Algorithm's termination will be based on number generation or when the chromosomes reach its maximum fitness value.



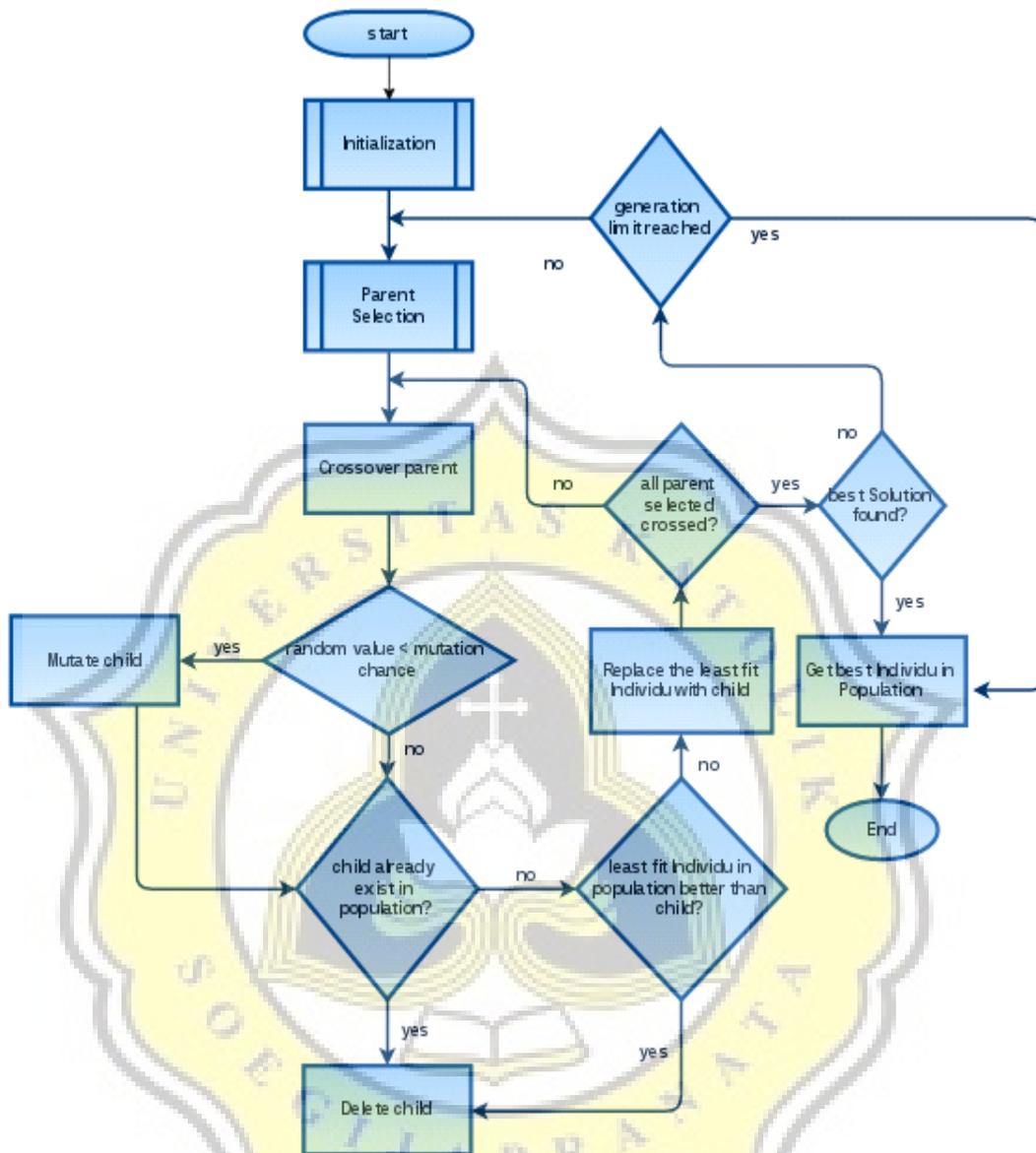


Figure 3. Flowchart Design of Genetic Algorithm

#### 4.2.2. Initialization

Initialization process here is to make population for the genetic algorithm. Figure 4 shows how the encoding in this project will work. The words that already stored in 2 dimension will be shuffled, then create a set of Boolean that will represent the direction of the words. The set of Boolean created by giving each value (T/F) 50% chance to show. If there are 5 words, it means there will be 5 Boolean values.

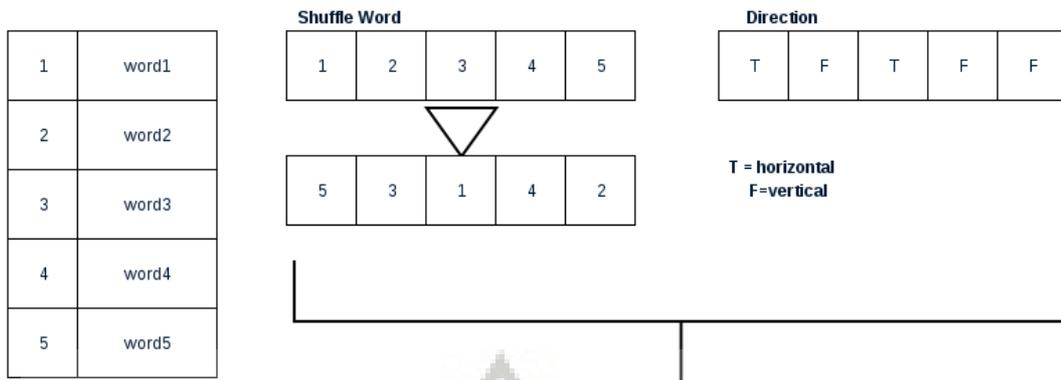


Figure 4. Encoding Set of Words and Set of Boolean

These two set of data will be combined and will become a chromosome (see figure 5) this process will be repeated until the population filled by chromosomes or individuals.

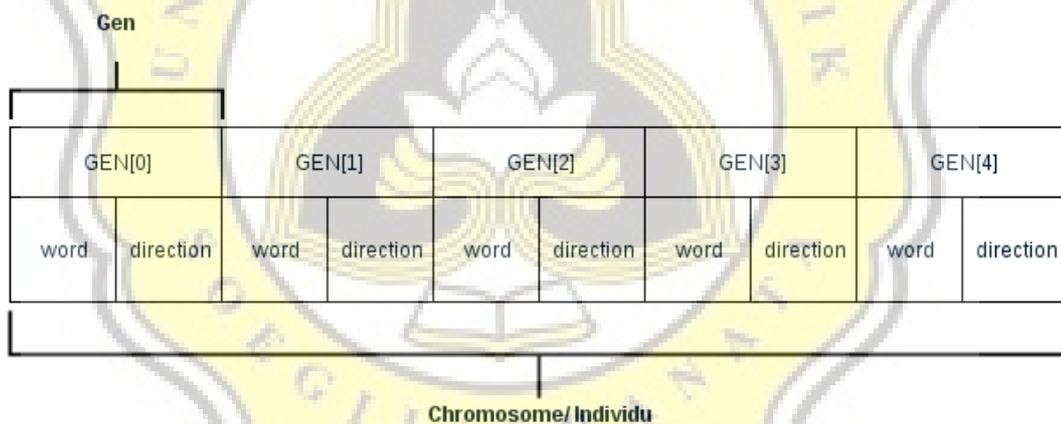


Figure 5. combine two arrays

#### 4.2.3. Count fitness

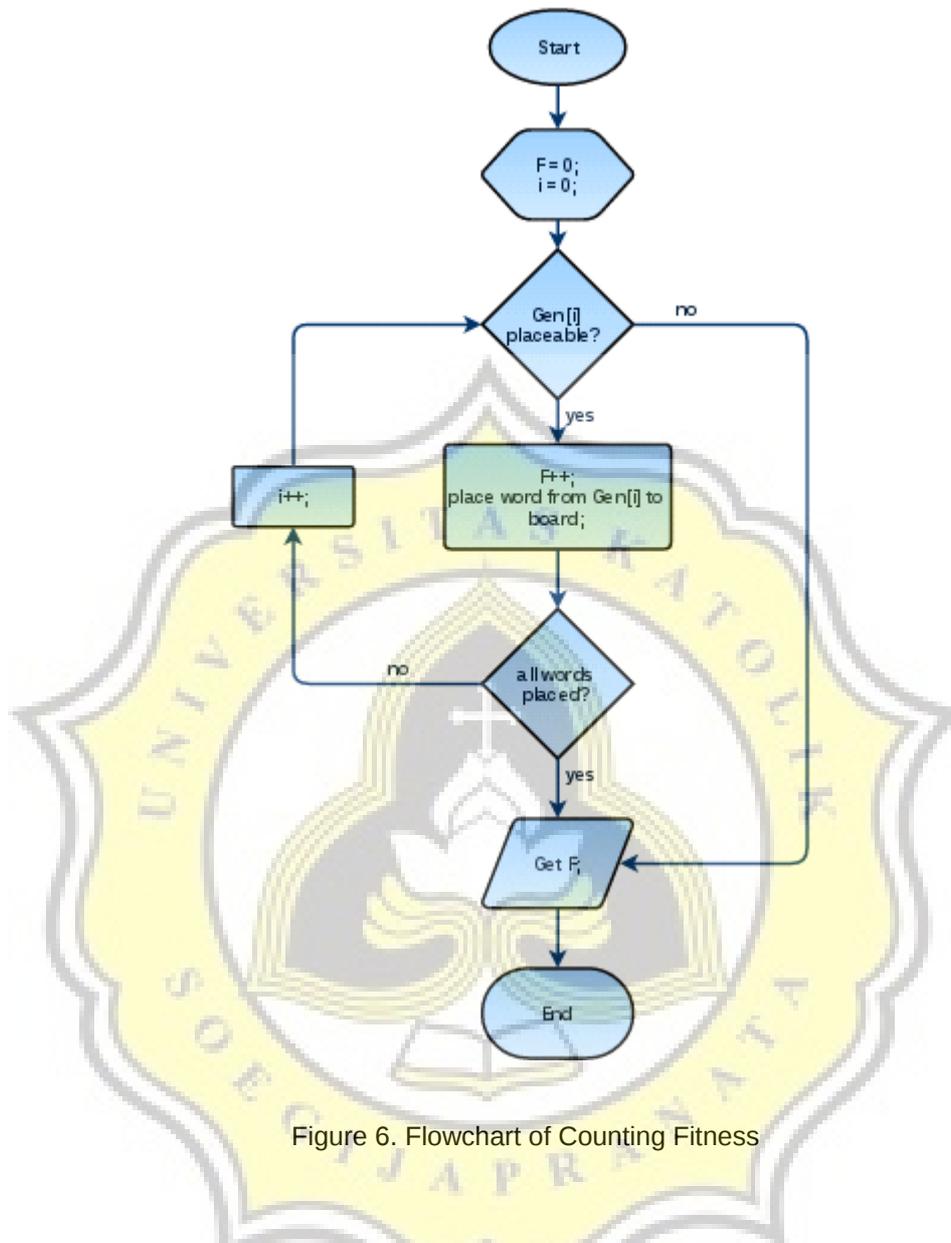


Figure 6. Flowchart of Counting Fitness

In this program, the fitness value of the chromosome based on how many word (gen) can be placed in a board. The program will try to place the word in each gene sequentially. Because solution used for this program are ordered-based. Example the value of a chromosome that contains word “hello” in the 3rd place and the same word in the 1st place will be different. So whenever a word placed in the board, the fitness will increase by one. But once a word can't be placed in the board, the

counting will be stop so the next word will not get a chance to be placed in the board.

#### 4.2.4. Selection

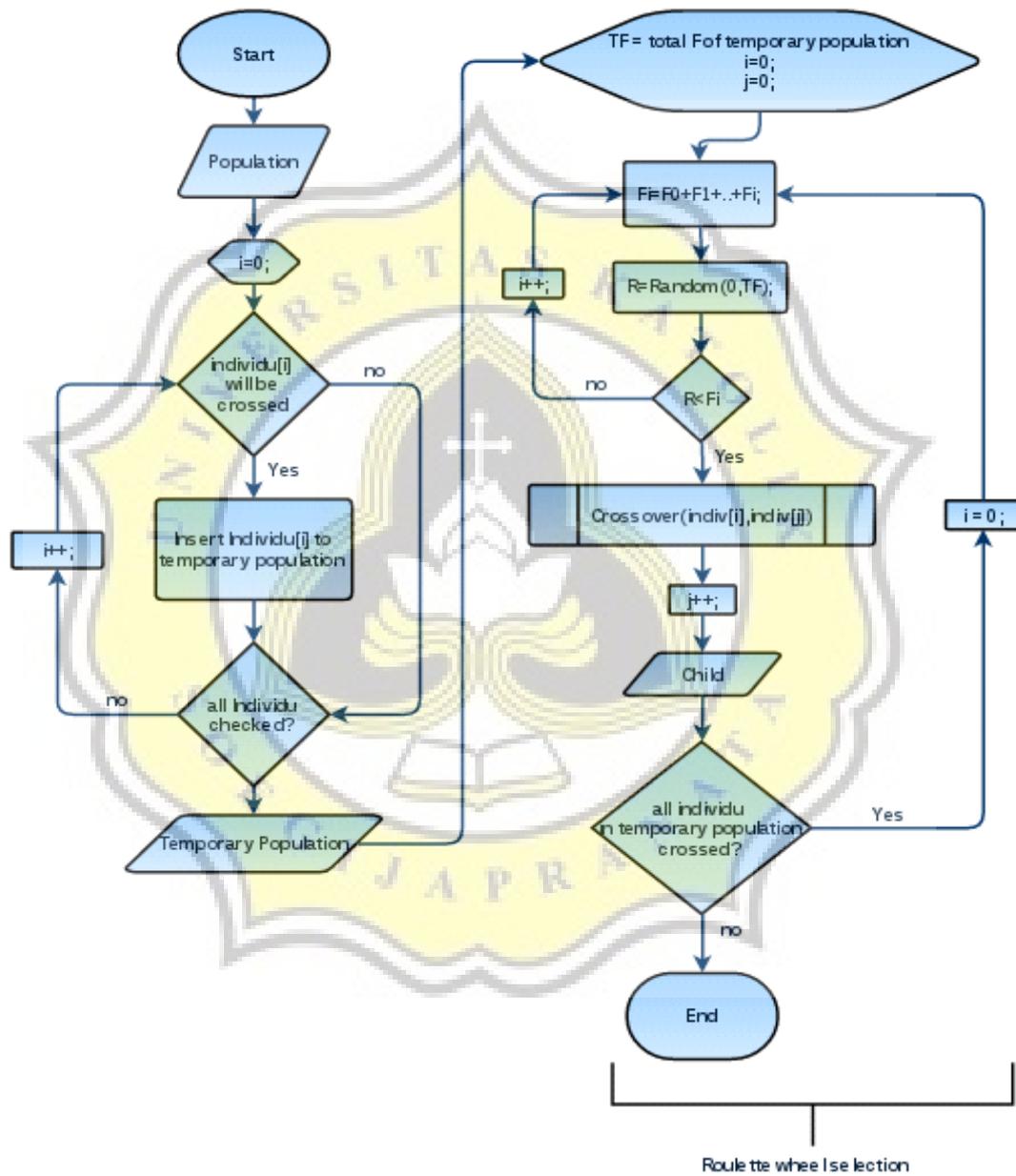


Figure 7. Flowchart of Selection process

This Process will be used to select two parents for the crossover process. Each of the chromosomes will have 80% chance whether it will be crossover. Each chromosome that pass the 80% chance will be stored in temporary population. Each of the chromosomes in temporary population then will be crossed over with another chromosome in the same temporary population. Roulette wheel selection used to choose that another chromosome. Roulette wheel selection chooses a chromosome like in a roulette wheel game. First the program will set some percentage to each chromosome. The total of percentage should be 100%. The percentage will depend on the fitness value. It means higher fitness value will have higher percentage. This roulette wheel selection will repeated until all chromosomes in temporary population have its partner to cross over

#### 4.2.5. Crossover

This crossover process is using single point crossover with ordered solution. Ordered solution mean even though two chromosomes have same value in its gen if both value located in different index of the chromosome it will have different meaning for the solution. Example value 5 at index 0 in chromosome1 will have different meaning with a chromosome2 that have value 5 at index 1. In this project, in a chromosome must not have two or more gen that have same value.

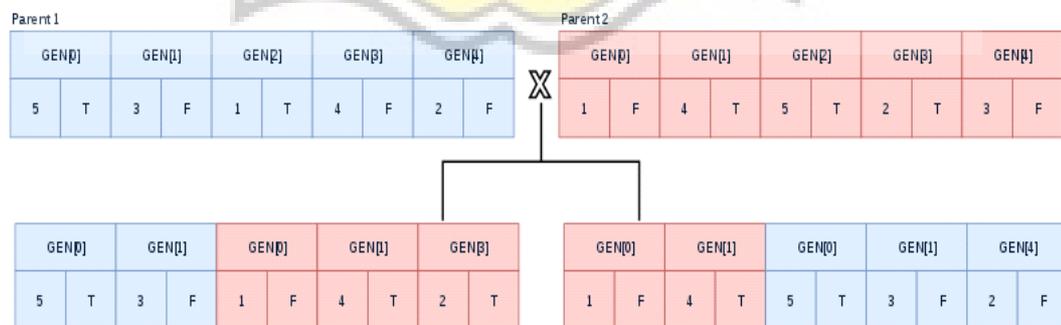


Figure 8. Crossover Process

At First the crossover will set a certain point where the chromosome will be split, in this program the point will be in the middle of the chromosome. Like in the image above, the child1 will have parent1's gene from gene [0] to gene [point]. Then for the rest of the gene, child1 will take parent2's gene that child1 does not have. This crossover will overcome 2 problems that stated before.

#### 4.2.6. Mutation

In this project user will input certain percentage of mutation. This mutation chance will decide whether a chromosome will mutate or not. The mutation used here is by reshuffle the words from the original chromosome. The reshuffled words then will be made into new chromosomes.

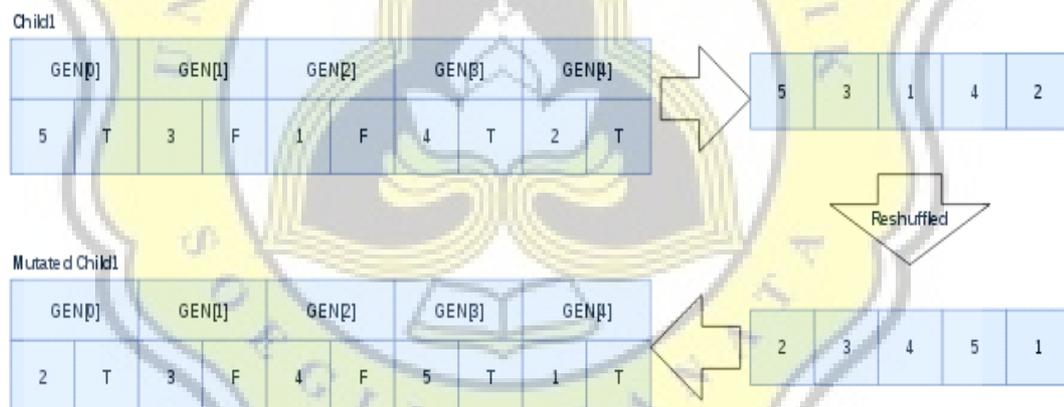


Figure 9. Mutation Process

#### 4.2.7. Class Diagram

In the program there are 15 java classes. But there are only 7 classes for the main algorithm. The rest of the classes are for the interface. Figure 10 shows the class diagram for those 7 classes. In class gen there are only 2 attributes in it. This class Gen used in class Individu. In class Individu there are array of Gen. Class Individu like its name it

purpose is for object Individu in the algorithm. In class Genetika there are array of object Individu. This is for the population. And Array list of object Individu is for the temporary pool for the selection process. Class Huruf is a node for linked list that will be used in class ListKata. And the class HitungFitness there will be array of object ListKata. This is for making an array of list. And for the last class, there are class Request. This object will be passed to other class because it is the object that stores all of the input and the result.

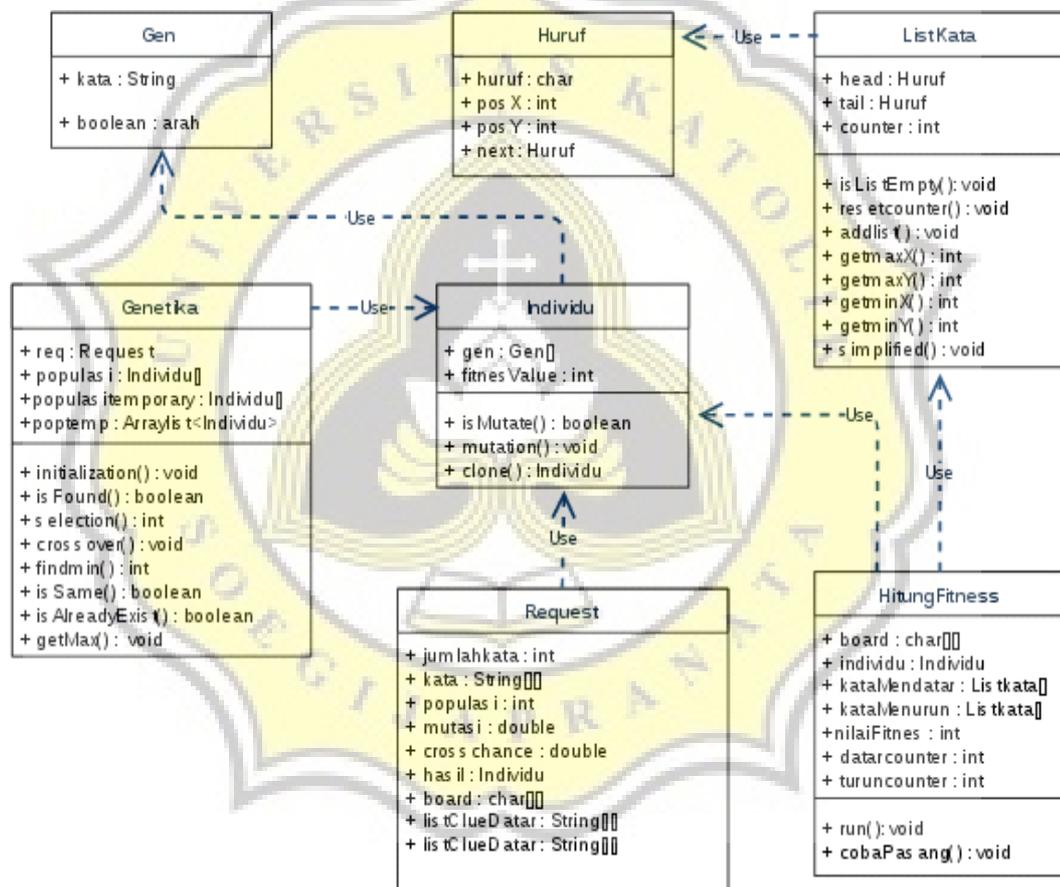


Figure 10. Class Diagram