CHAPTER V
IMPLEMENTATION & TESTING

5.1 Implementation

This sub chapter will explain the program logic and interface. This program use Graphic User Interface (GUI) for ease of use.

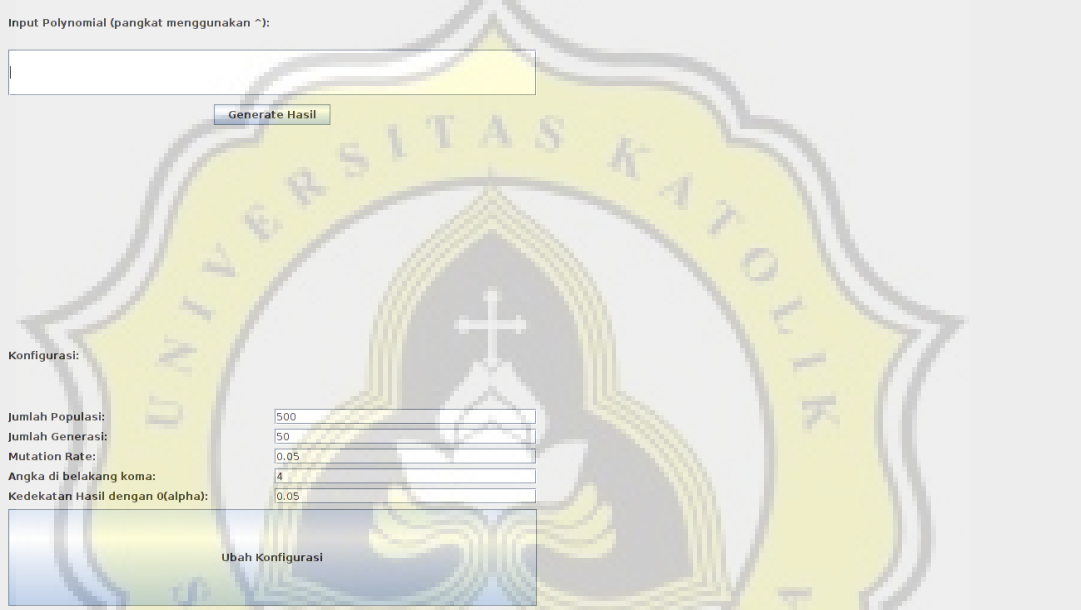

Figure 3. Graphic User Interface before use

The left side is a part of program that the user will input the polynomial problem (on upper left) and change the configuration (on lower left). The right side will display the graph after GA process is done.

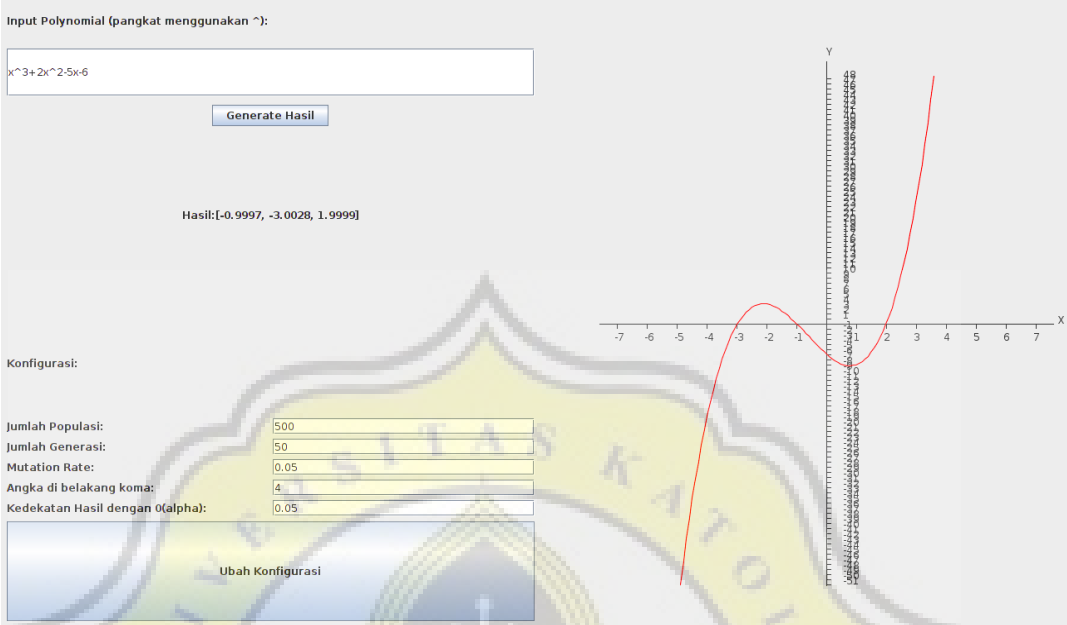

Figure 4. Graphic user interface after calculation is done

Input from the user is a polynomial function in a string form, but for the calculation program needs to read the constant in numerics. So, the program have to divide and convert it into array of numerics. Program will divide the string based on the sign. Example: for 2x^2+3x+1, program will divide it into 2x^2 , 3x , and 1. After this step, program will group the exponents and constants into separate arrays. The array of constant will contain 2, 3, and 1. The array of exponent will contain 2, 1, and 0. The sequence of the arrays will show the polynomial terms. After this process, program will get the bound of the polynomial roots. The bound will be used for the range to generate the population.

For generating population, program divides the range into parts. Then, program picks the number randomly from between each part of the range to insert the number into the population. After that, each of population is evaluated by the program. The population will subtitute x value of the polynomial equation. Program uses the conversion of the equation to calculate the function value. The function value is the resut of the evaluation process.

Fitness value is produced from the reciprocal of the evaluation result. After program get fitness value, program will input the fitness value into roulette wheel for selection process.

Selection process will select better solutions and eliminate bad solutions from population. Then, bad solutions will be replaced with the result of crossover process. Mutation process will be done after crossover process. Mutation process only occur on small chance (between 1-10%). Mutation process will switch one of the gene into different value and produce new population.

Program will loop from evaluate population until mutation process. Each loop will generate new population. Loop will continue until all the roots are found or the maximum number of generation is reached. Program will stop the process and display the best solutions (if program find the solutions) along with the cartesian graph.

5.2 Testing

This section will show testing of the program if the config is changed below recommended number. The recommended number for the population is above 100, and for the generation is above 10.

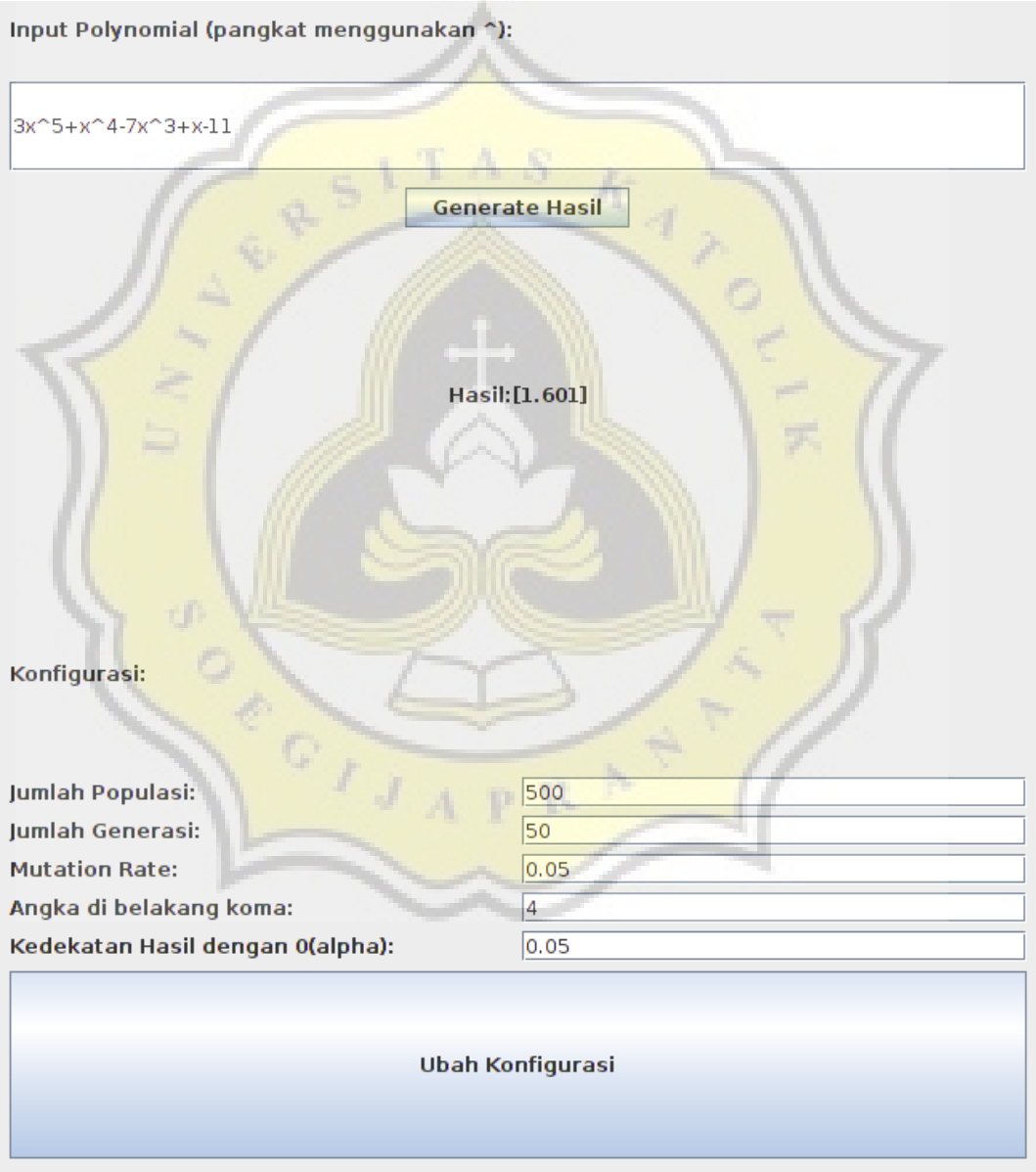

Figure 5. Result before testing

For the first experiment, the population will be reduced to number below suggestion (below 100). The solution is not found.

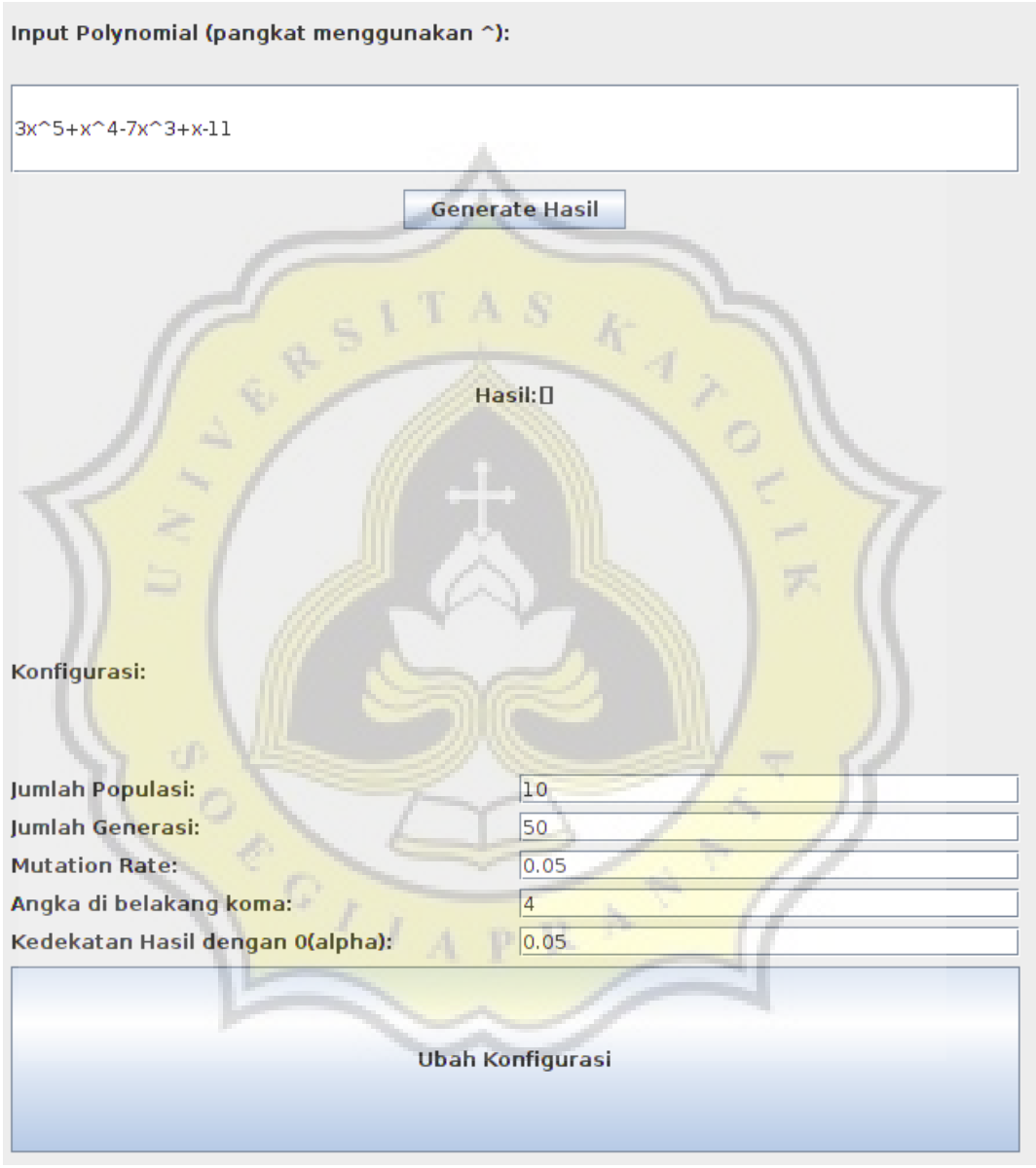

Figure 6. First Experiment

For the second experiment, generation will be reduced to number below suggestion (below 10). The solution is not found.

Figure 7. Second Experiment

For the third experiment, the decimal accuracy and alpha(closer result to zero) is changed. If user increase the decimal number and lower the alpha, the accuracy of the solution is increased. However, the population and the generation should have recommended number (above 100 for population, above 10 for generation) because it will affect the final result.

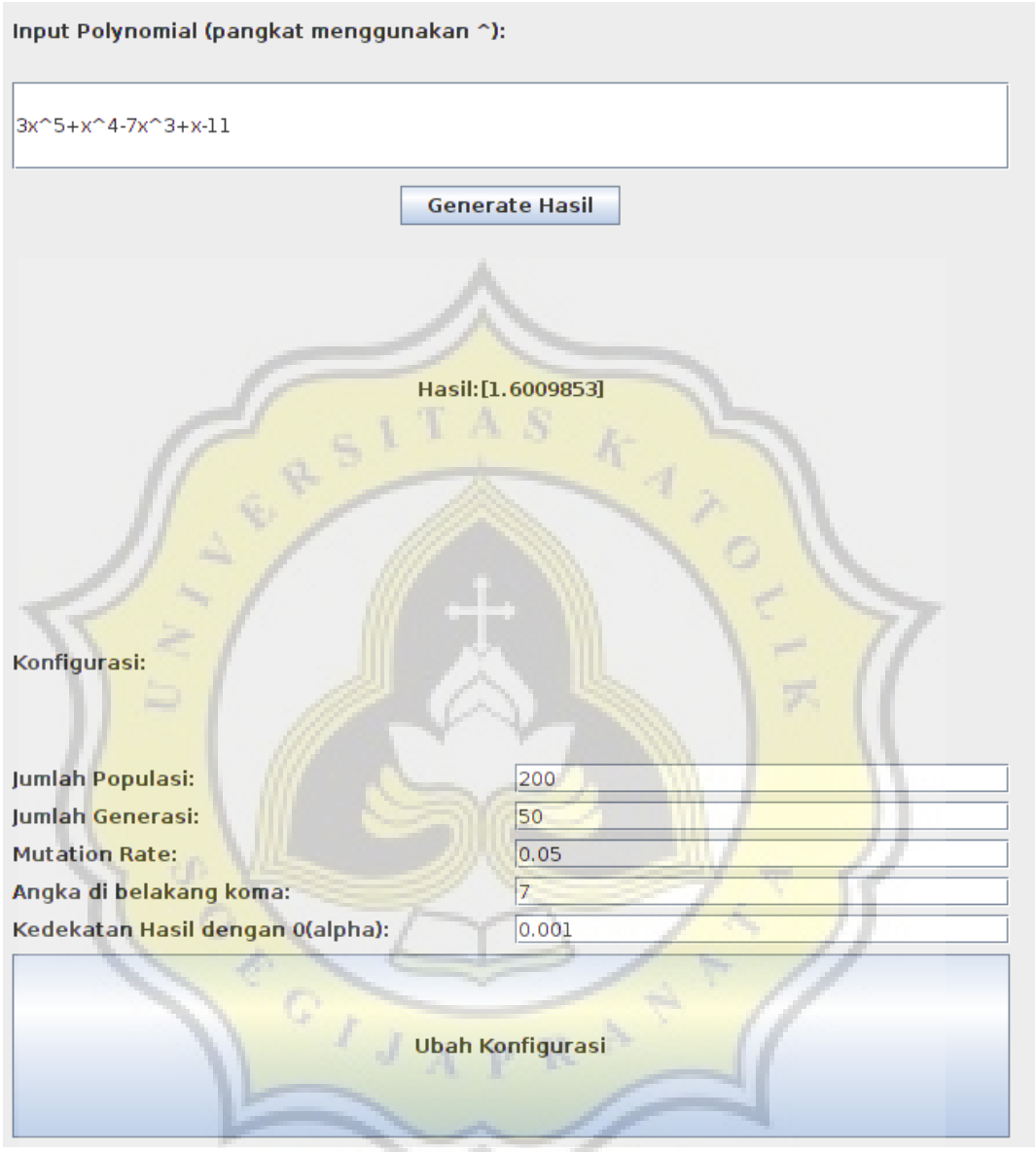

Figure 8. Third Experiment