Implementation of Genetic Algorithm in Determining Class Schedules Based on User Needs

Agus Prayogo [a1], I Gede Santi Astawa[a2], I Gusti Agung Gede Arya Kadyanan[a3], Ida Bagus Gede Dwidasmara[a4], Dr. Ngurah Agus Sanjaya[a5], Ida Bagus Made Mahendra[a6]

[a]Informatics Department, Faculty of Mathematics and Natural Sciences, University of Udayana
South Kuta, Badung, Bali, Indonesia
[1]agusprayogo99@gmail.com
[2]ibm.mahendra@unud.ac.id
[3]agus_sanjaya@unud.ac.id
[4]santi.astawa@unud.ac.id
[5]dwidasmara@unud.ac.id
[6]gungde@unud.ac.id

Abstract

Education is an important thing in life. In practice, education consists of teachers and students. A teacher teaches a subject to students. Students are taught by several or more teachers. There is a provision in which a class of students can only receive one subject at a time. Teachers can also only teach one subject at a time. This requires a system that is used to regulate so that these provisions can be fulfilled without ignoring other provisions. The system used to manage these problems is a subject scheduling system. This system regulates the class placement of students with the subjects they study. In manual implementation, the process is very inefficient in terms of time, human resources and thought power. The method that can be used to solve this problem is the genetic algorithm. A genetic algorithm is a heuristic method or procedure inspired by the natural selection process. Genetic algorithms are generally used to produce high-quality solutions to problems in the form of optimization and search by relying on biologically inspired operators such as mutation, crossbreeding, and selection

Keywords: *Education, Scheduling, Genetic Algorithm, Optimization,*

1. Introduction

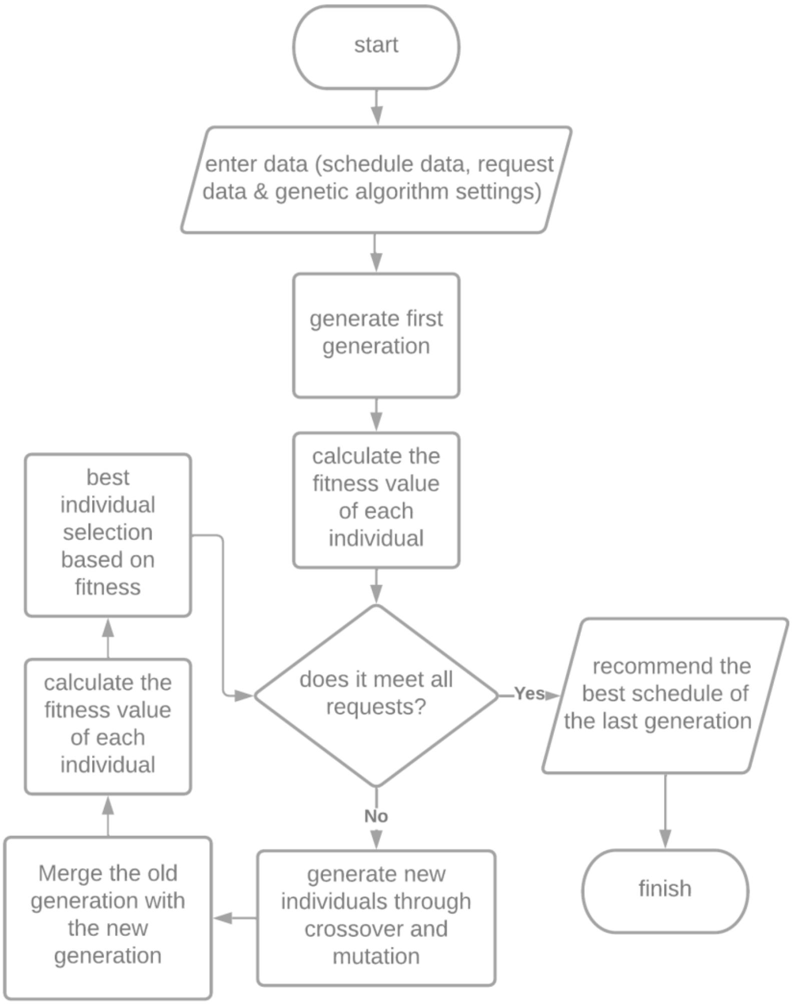
Education is an important thing in life. There are several institutions that are trusted in creating quality education, such as schools and colleges. The education is organized in a structured and complex system. The system consists of smaller systems, where the small system consists of various components that work according to their respective roles and capacities. These components complement each other, so the system will not be able to work properly if one of them is disturbed. In practice, education consists of teachers and students. A teacher teaches a subject to students. Students are taught by several or more teachers. There is a provision in which a class of students can only receive one subject at a time. Teachers can also only teach one subject at a time. This requires a system that is used to regulate so that these provisions can be fulfilled without ignoring other provisions. The system used to manage these problems is a subject scheduling system. This system regulates the class placement of students with the subjects they study. This system has problems, such as colliding schedules, teachers who cannot teach at a certain time, and so on. In manual implementation, the process is very inefficient in terms of time, human resources and thought power. If a predetermined schedule cannot meet all applicable provisions, then the schedule cannot be used in teaching activities. The schedule needs to be reworked. For a small scale, creating a schedule manually is quite easy. However, for a larger scale that is generally used in large institutions that require greater flexibility, it requires system that can solve these problems quickly and precisely. For that, a method is needed to build the system. The method that can be used to solve this problem is the genetic algorithm. A genetic algorithm is a heuristic method or procedure inspired by the natural selection process. Genetic algorithms are generally used to produce high-quality solutions to problems in the form of optimization and search by relying on biologically inspired operators such as mutation, crossbreeding, and selection. The genetic algorithm has several variables that can affect the optimal level of the results obtained. Some of these variables are the percentage of mutations and the percentage of crossover. This research is expected to find the best percentage of mutations and percentage of crossover in order to get the most optimal results. Several research have succeeded in using genetic algorithms in making scheduling systems. One of them is [1] research with the title "Implementation of Genetic Algorithms in Web-Based Lecture Scheduling Applications by Adopting the Waterfall Model". At the conclusion of it, the system he created was successful in carrying out the scheduling process quickly without any conflicting schedules. The application can also reset the schedule if there are lecturers who cannot teach at certain times. However, the system cannot add requests given by system entities. Thus, the system only relies on changes to the final schedule. In this research, the author adds a new feature where users can add new conditions to the system where the system must meet these conditions in order to get the desired results.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

2. Research Methods

2.1 Genetic Algorithm

The following is the design of the Genetic Algorithm method in Determining Lecture Schedules Based on User Needs


start
enter data (schedule data, request data & genetic algorithm settings)
generate first generation
calculate the fitness value of each individual
does it meet all requests?
Yes
recommend the best schedule of the last generation
finish
No
generate new individuals through crossover and mutation
Merge the old generation with the new generation
calculate the fitness value of each individual
best individual selection based on fitness

The following is a description of the flow chart above.

1. In the first step, the user enters the data namely;
    a. Schedule data, namely data containing a list of teaching classes, a list of teachers, a list of courses, a list of activities, a list of classrooms, working days, and the number of teaching hours each day
    b. Request data
    c. Data on the number of individuals per generation.
2. Then, the system will generate the first schedule of a number of individuals per generation.
3. The fitness of that generation will be calculated to determine the best schedule. Fitness is obtained from 2 steps, namely;
    a. Counting the number of activities that collide with each other, where a teacher or a teaching class is assigned to 2 or more rooms at the same time. The fewer the number of collisions, the better.
    b. Calculation of the match of requests with those obtained in the schedule. The more requests a schedule fulfills, the better.

4. If from that generation there is not a single schedule that can fulfill all requests, then go to step 5. If so, go to step 9.
5. Production of new generations from previous generations through mutation and crossover processes.
6. From the new generation, combine with the existing generation to become a larger population.
7. Calculate the fitness of the large population as done in step 3
8. Prune the population by taking only the best individuals of the initial generation. Go back to step 4.
9. Show the best choice of schedule recommendations.

2.2 Scheduling

Scheduling comes from the word schedule which gets the suffix pen which means time division based on a work order arrangement plan or list or activity table or activity plan with a detailed implementation time division. Scheduling is a very important problem in an educational institution [1].

3. Result and Discussion

3.1. Database Schema

the following is the schematic of the base system

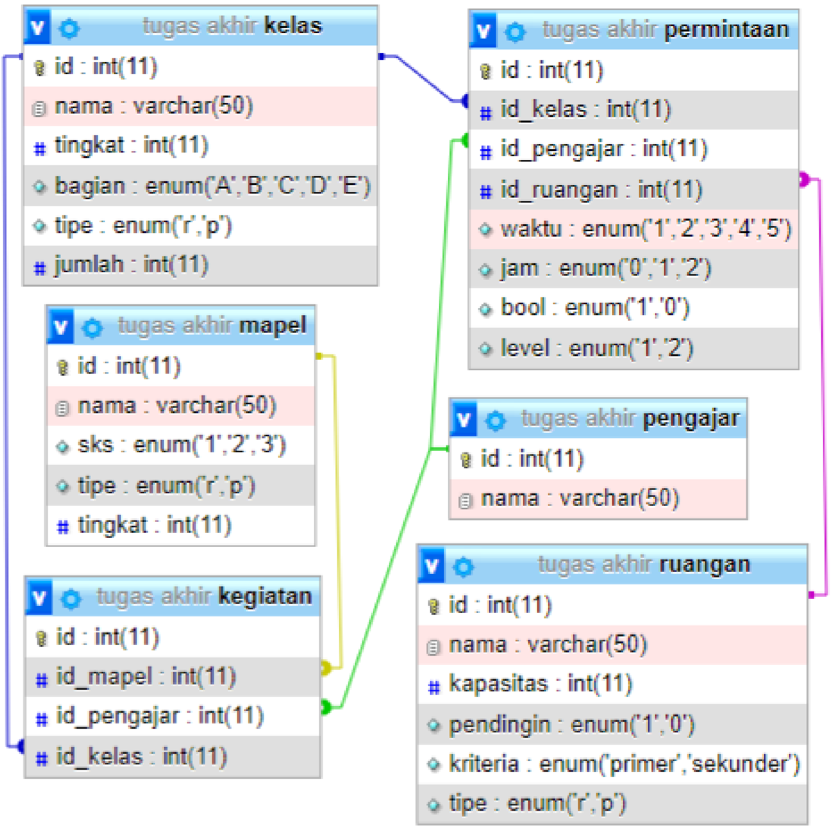
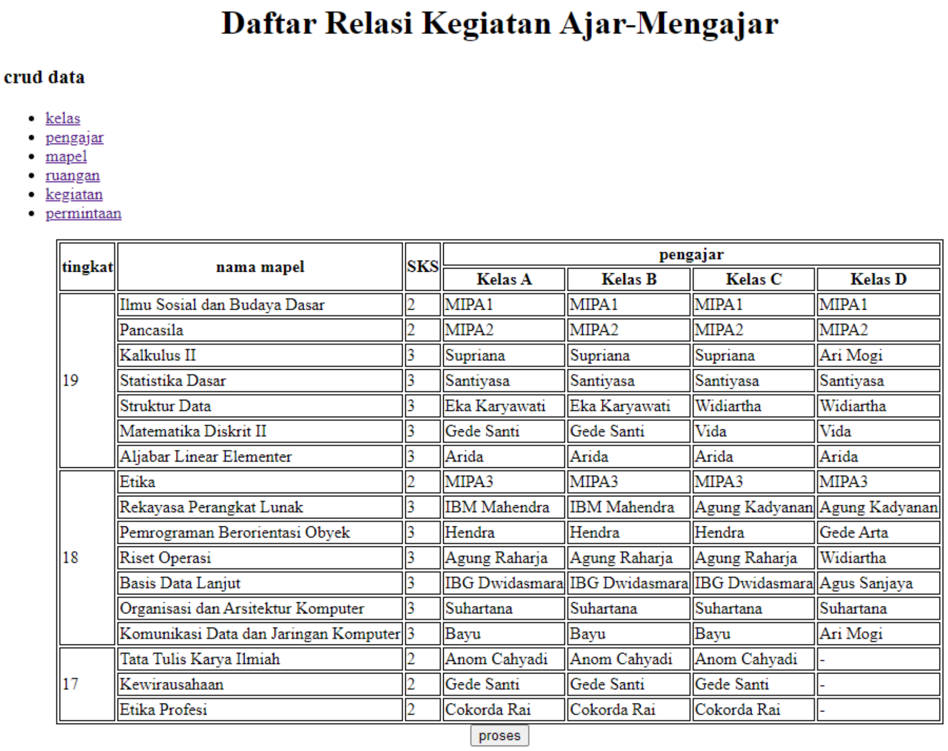


Figure 3. system database schema

system database consists of 6 tables, namely:
- *kelas*, that is, student class
- *permintaan*, the requests from teachers and student classes
- *mapel*, that is lecture subjects
- *pengajar*, that is lecturer
- *kegiatan*, namely the relationship between lecturers, subjects, and classes
- *kelas*, namely the classroom where the activity is carried out

3.2. System Interface Design

The system for determining the lecture schedule is implemented on a web-based basis. The following is the design of the system interface.



Daftar Relasi Kegiatan Ajar-Mengajar

crud data

- kelas
- pengajar
- mapel
- ruangan
- kegiatan
- permintaan

tingkat	nama mapel	SKS	pengajar			
			Kelas A	Kelas B	Kelas C	Kelas D
19	Ilmu Sosial dan Budaya Dasar	2	MIPA1	MIPA1	MIPA1	MIPA1
	Pancasila	2	MIPA2	MIPA2	MIPA2	MIPA2
	Kalkulus II	3	Supriana	Supriana	Supriana	Ari Mogi
	Statistika Dasar	3	Santiyasa	Santiyasa	Santiyasa	Santiyasa
	Struktur Data	3	Eka Karyawati	Eka Karyawati	Widiartha	Widiartha
	Matematika Diskrit II	3	Gede Santi	Gede Santi	Vida	Vida
	Aljabar Linear Elementer	3	Arida	Arida	Arida	Arida
18	Etika	2	MIPA3	MIPA3	MIPA3	MIPA3
	Rekayasa Perangkat Lunak	3	IBM Mahendra	IBM Mahendra	Agung Kadyanan	Agung Kadyanan
	Pemrograman Berorientasi Obyek	3	Hendra	Hendra	Hendra	Gede Arta
	Riset Operasi	3	Agung Raharja	Agung Raharja	Agung Raharja	Widiartha
	Basis Data Lanjut	3	IBG Dwidasmara	IBG Dwidasmara	IBG Dwidasmara	Agus Sanjaya
	Organisasi dan Arsitektur Komputer	3	Suhartana	Suhartana	Suhartana	Suhartana
	Komunikasi Data dan Jaringan Komputer	3	Bayu	Bayu	Bayu	Ari Mogi
17	Tata Tulis Karya Ilmiah	2	Anom Cahyadi	Anom Cahyadi	Anom Cahyadi	-
	Kewirausahaan	2	Gede Santi	Gede Santi	Gede Santi	-
	Etika Profesi	2	Cokorda Rai	Cokorda Rai	Cokorda Rai	-

Figure 4. System interface main page design

Figure 4 is the main view of the system. On this page the user can see all the data systematically, where the data is presented in the form of a relationship between one data and another to make it easier for the user to manage the data.

There is also the system can save the table that was successfully formed in .xls format using Javascript programming. Here is a snippet of the program code syntax.

Table 1. The syntax snippet of the function exports a table in the form of .xls

```
-  function exportTableToExcel(tableID, filename = ''){
-      var downloadLink;
-      var dataType = 'application/vnd.ms-excel';
-      var tableSelect = document.getElementById(tableID);
-      var  tableHTML  =  tableSelect.outerHTML.replace(/   /g,
    '%20');
-      // Specify file name
-      filename = filename?filename+'.xls':'excel_data.xls';
-      // Create download link element
-      downloadLink = document.createElement("a");
-      document.body.appendChild(downloadLink);
```

3.3. Genetic Algorithm Implementation

1. Initialization

The first step in the genetic algorithm is to generate a number of individuals that are used as the first generation. The types of data used in constructing the individual are activity data and room data.

2. Evaluation

From a number of individuals in the first generation, the fitness value of each generation is calculated. The fitness value is the value to determine the level of goodness of the individual. In this study, the researcher determined that the higher the fitness value of an individual, the better the individual.

3. Reproduction

Reproduction is done in 2 ways, namely crossover and mutation. The crossover method used is Partially Mapped Crossover (PMX). PMX is a 2-point crossover with the addition that no chromosome in an individual can exist more than once.

4. Reevaluation

New individuals resulting from the reproduction process will be calculated to obtain a fitness value to continue in the elimination process.

5. Elimination

The elimination method used is the Roulette Wheel. In this process, individuals who have good fitness values will have a greater chance of being selected as individuals in the next generation.

3.4 Testing

Here are the results of the tests that have been carried out. The graphs below display the fitness value of the best individual in a generation in 4 experiments on the system. In the graph, the vertical axis indicates the value of the best individual in a generation, while the horizontal is the order of the generations. Each experiment was carried out with repetitions of 100 generations.

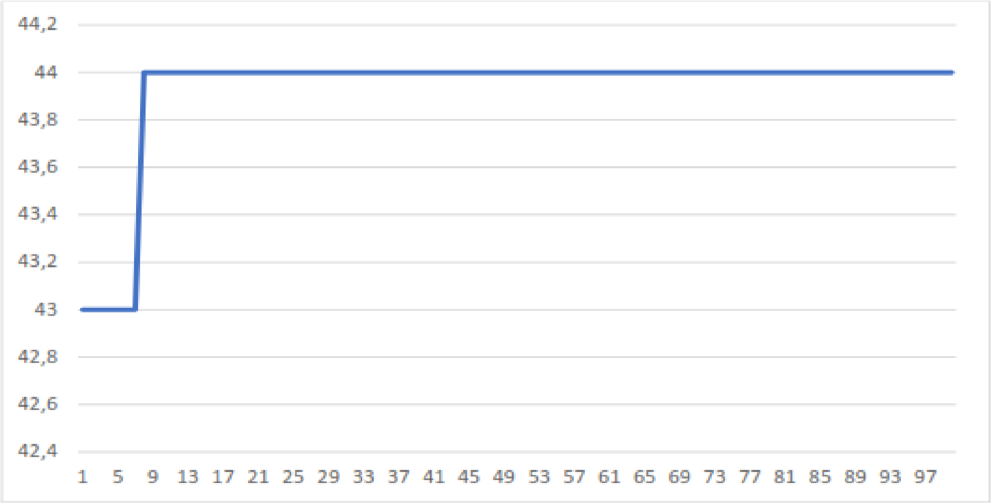


Figure 5. The graph of the best individual change on the first try

p-ISSN: 2301-5373
e-ISSN: 2654-5101

From figure 5 above, first generation has greatest value of 43 constantly until 7th generation, then increased to 44 at 8th to 100th generation.

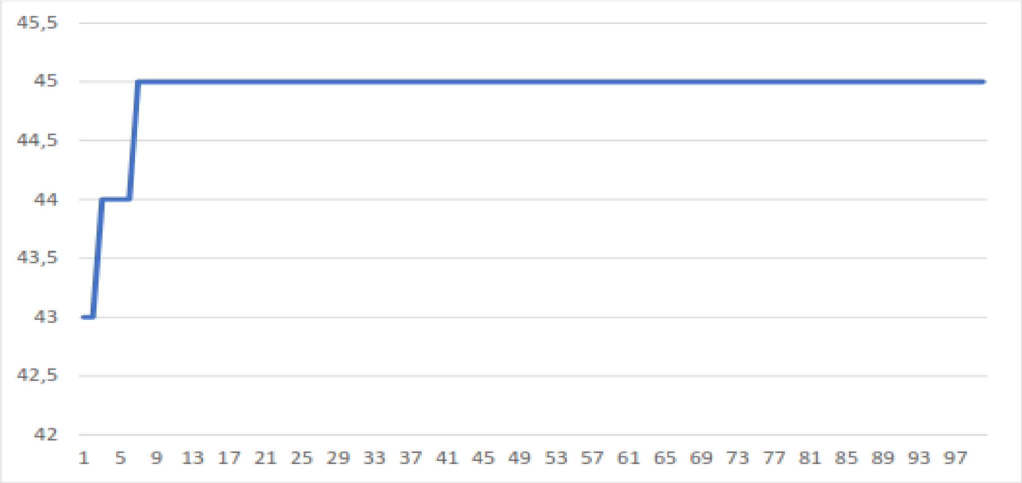


Figure 6. The graph of the best individual change on the second tries

On the next figure, first generation has greatest value of 43 to 2nd generation, then increased to 44 at 3rd to 6th generation, then increased again to 45 at 7th to 100th generation.

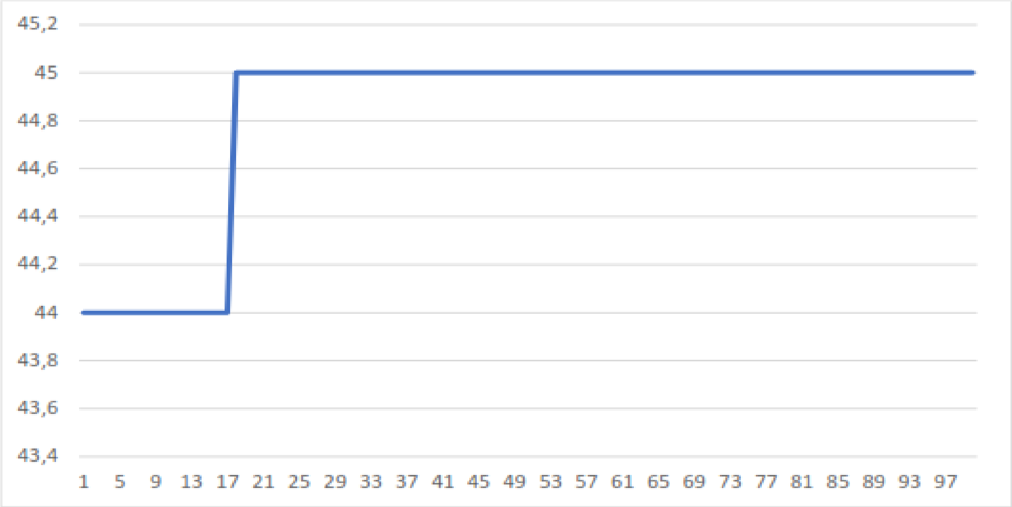


Figure 7. The graph of the best individual change on the third tries

Next one, the first generation has greatest value of 44 to 17th generation, then increased to 45 at 18th to 100th generation.

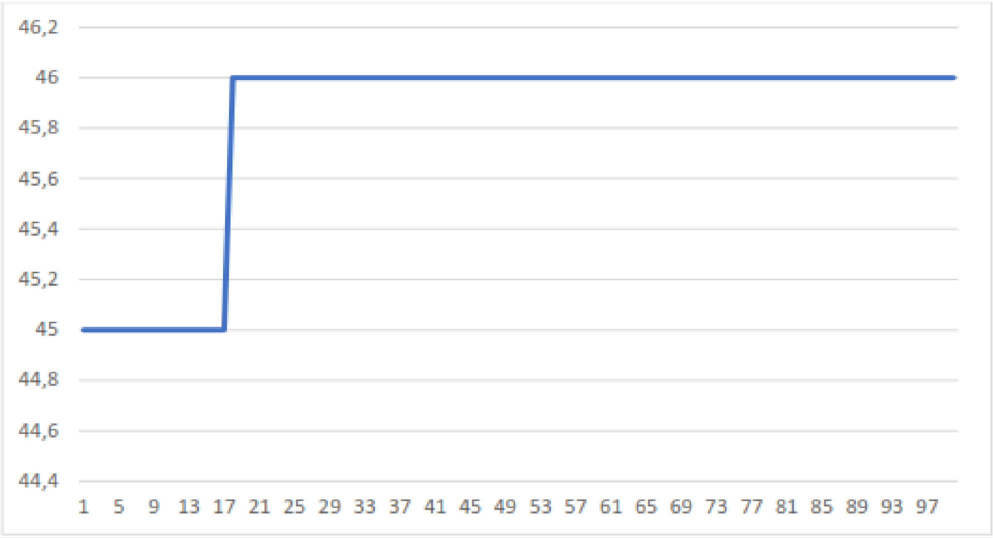


Figure 8. The graph of the best individual change on the 4th tries

Lastly, the first generation has greatest value of 45 to 17th generation, then increased to 46 at 18th to 100th generation.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

From the graphs above, it can be concluded that the average increase in the fitness value of the best individuals in each generation is:

- trial 1 : 0.010101
- Trial 2 : 0.020202
- Trial 3 : 0.010101
- Trial 4 : 0.010101

For the calculation of accuracy, the 5th experiment was carried out with all the same parameters as the previous experiment. In the 100th iteration, the individual with the best fitness value is 45. The individual has a total of 12 activities that have requests, with 7 of them being fulfilled, and 96 possible collisions with no collisions. From the data obtained, the accuracy can be calculated as follows.

$$accuration = \left(0.5\frac{the\ total\ number\ of\ collisions\ that\ occurred - number\ of\ collisions}{Total\ number\ of\ collisions\ that\ can\ occur}\right) + \left(0.5\frac{the\ number\ of\ activities\ that\ meet\ the\ demand}{the\ total\ number\ of\ activities\ that\ have\ a\ demand}\right)$$

$$= \left(0.5\frac{96-0}{96}\right) + \left(0.5\frac{7}{12}\right) = \left(0.5\frac{96-0}{96}\right) + \left(0.5\frac{7}{12}\right) = 0.79166$$

4. Conclusion

From the research that has been done, the conclusions obtained is, the genetic algorithm has succeeded in optimizing the predetermined schedule. However, it never got the best results even though it had been done for 100 generations due to the shortcomings of the genetic algorithm. It takes at least 18 iterations to get a constant value with increments as high as 1-2 fitness values The system has an accuracy of 0.7966 with a value of 1 as a form of perfect accuracy.

References

[1] Ahmat Josi. *Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall (Studi Kasus STMIK Prabumulih)*. 2017

[2] I Made Suwija Putra. *Penerapan Algoritma Genetika Dan Implementasi Dalam Matlab*. 2018

[3] Tanweer Alam, et al., *Genetic Algorithm: Reviews, Implementations and Applications.* 2020

[4] Listanto Tri Utomo, Farid Wahyudi. *Implementation Of Genetic Algorithm in Optimization of Subject Scheduling.* 2020

[5] David Kristiadi, Rudy Hartanto. *Genetic Algorithm for Lecturing Schedule Optimization (Case Study: University of Boyolali). 2019*

This page is intentionally left blank