Telkom Bandung vocational school scheduling application using a website-based genetic algorithm

Putri Ramadhani[1], Alfian Nasir[1], Zakia Mahbub Abdullah[1], Astri Novianty[1], Casi Setianingsih[1]
[1]Department of Computer Engineering, School of Electrical Engineering, Telkom University, Indonesia

Article Info

Article history:

Received September 9, 2023
Revised September 12, 2023
Accepted September 13, 2023

Keywords:

Scheduling,
Genetic Algorithm

ABSTRACT

Scheduling is a very important factor for the learning process at school, it is also very important at SMK Telkom Bandung. At this school, the schedule is still done manually. Because of this, many problems occurred, there were schedule clashes, and the teaching and learning process was hampered. Therefore, a scheduling system was created using a genetic algorithm which is one of the optimization algorithms and can be used in various case studies such as scheduling subjects at school. With the application of the genetic algorithm, it can produce an automatic and optimal subject schedule. From several trials on the genetic algorithm, the best fitness value is 1 with an average execution time of 14.42990657 seconds.

Corresponding Author:

Alfian Nasir
Department of Computer Engineering
School of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: alfnnsr@student.telkomuniversity.ac.id

1. INTRODUCTION

Telkom Bandung Vocational High School (SMK) is a vocational school under the auspices of the Telkom Education Foundation (YPT) which was founded in 2013. The school has three Study Programs, namely Telecommunications Access Network Engineering, Computer and Network Engineering, and Multimedia. Based on the five existing study programs, there are several classes. The number of classes follows the number of study program enthusiasts at each new student admission. This causes the number of study programs for class X, XI and XII is not the same. In addition, Telkom Bandung Vocational School has a curriculum system that is different from SMKs in general, where there are odd and even weeks. On odd weeks class XII carries out productive methods, namely doing all practicums. While grades X and XI carry out non-productive methods, namely taking general subject learning, and the following week the methods were switched. With the existing curriculum system, the scheduling of subjects at SMK Telkom Bandung is a complex problem and often experiences difficulties. This is due to several factors that must be considered, including the number of classes used, the number of subjects available, and some teachers who cannot teach at certain hours which often becomes a problem in the scheduling process. The design of the wrong scheduling system can cause problems for both teachers and students [1]. Some of the problems that arise include conflicts caused by clashes between subjects and classes in one schedule. To overcome this, an appropriate algorithm is needed to design a scheduling system. Of the several existing algorithms, a number of studies use genetic algorithms to solve scheduling problems. Genetic algorithms are suitable for complex cases and produce better output than other algorithms [2].

Genetic Algorithms are actually inspired by the principles of genetics and natural selection (Darwin's theory). Discovered at the University of Michigan, United States by John Holland through research and popularized by one of his students, David Goldberg, producing a book entitled "Adaptation in Natural and Artificial Systems" in 1975. The basic concept genetic algorithms are actually designed to detect processes in natural systems that are necessary for evolution, especially the theory of natural evolution proposed by Charles Darwin, namely survival of the fittest. According to this theory, in nature there is competition between individuals for scarce natural resources so that strong creatures dominate weak creatures [3]. The development of research using the Genetic Algorithm method has produced a better level of accuracy. Related research entitled Lecture Scheduling System Using Genetic Algorithms uses case studies in the Elementary School Education Study Program[4] Conduct research to optimize the subject schedule by explaining the availability of courses, rooms, implementation times, as well as the presence and availability of teaching staff.

So far, subject scheduling, especially in this research, a case study at Telkom Bandung Vocational School, which includes the division of subjects, classes and teachers, still uses manual methods, so there are several teacher schedules that conflict. To divide teachers according to classes and subjects given to teachers within a certain time requires quite complicated regulations and takes a long time to complete the scheduling. In preparing this subject schedule, there are many possibilities that should be tried to find the best scheduling. Therefore, an optimization method is needed that can be applied to work on scheduling these subjects, namely by using an artificial intelligence method, namely the Genetic Algorithm [5].

Therefore, a Scheduling Application for SMK Telkom Bandung was created using a Website-Based Genetic Algorithm so that it is able to provide output in the form of an automatic scheduling system that provides solutions to schedules that are out of sync between the needs of teachers and students. Schedules that have been arranged automatically provide convenience or solutions for schools to carry out learning hours in class without schedule conflicts occurring at the same time. The results of this study can be used as an alternative to automatic scheduling of subjects at the school level.

2. THEORETICAL BASIS

With the many study programs available at Telkom Bandung Vocational School, a curriculum is needed that can develop students to be able to learn independently, learn to be creative and also require innovation, so they can compete in the business and industrial world. Combining the national curriculum and innovation activities certainly requires special considerations. One of them is in arranging the subject schedule for each class. Based on the results of a survey conducted by the author at Vocational School Telkom Bandung, according to Omega Julyanto, S.E, M.M as part of the curriculum there are several problems/complaints in scheduling. There are, subject scheduling work at Telkom Bandung Vocational School is still done manually, there are still frequent conflicts in preparing subject schedules and the curriculum department wants to recreate the scheduling system, but this has not been realized.

Each school has different scheduling problems and produces different solutions. According to Ozturk and Sagir [6], in general, scheduling is an optimization problem managing time, space, and resources which are often limited simultaneously. This is motivated by the different boundaries used and the goals to be obtained. Consideration of various aspects such as the number of teachers, subjects, classrooms/labs and available time is also a problem in the subject scheduling process.

As for overcoming the scheduling problem can be done by using a genetic algorithm. According to Gregorius Satia Budhi, Kartika Gunadi and Denny Alexander Wibowo [7] Genetic Algorithm is a method for finding the optimal solution to a problem. This method will find the best solution by crossing the possibility of one solution and another to create a new solution. After that, the method will mutate the new solution into several parts of the solution offspring. The process will start by creating a random population of solutions/chromosomes. Then, the genetic algorithm will calculate the fitness value of each chromosome in the population. Next, two chromosomes will be selected to be crossed and produce offspring. Then the offspring will mutate. This process will continue to repeat until the best conditions are reached. Ihab Sbeity, Mohamed Dbouk and Habib Kobeiss [8], combined the Analytical Hierarchy Process (AHP) with a genetic algorithm to complete the scheduling process in schools. AHP is used to assess teachers according to the criteria provided by the school. This value will be input for the genetic algorithm in determining the schedule according to the given limits and according to the wishes of the teachers.

3. METHOD

3.1 Genetic Algorithm

The Genetic Algorithm has variables and parameters, each individual has a fitness function (objective function) to be able to determine the level of suitability of the resulting output, the population of the number of individuals included for each generation, the possibility of crossover in generations, mutations in individuals and the number of generations made that determines genetic algorithm[9] [10].

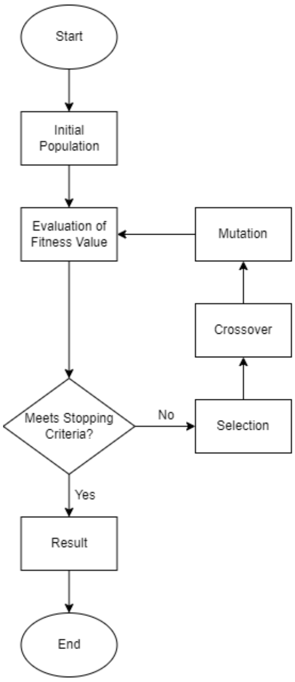


Figure 1. Genetic Algorithm Flowchart

a. Population initialization

Population initialization is a technique in the implementation of the earliest genetic algorithms. This technique initializes the randomized chromosomes into the population [11]. The system's initial population consists of five genes: teacher, subject, room, hour, and day. For more details, see the following image.

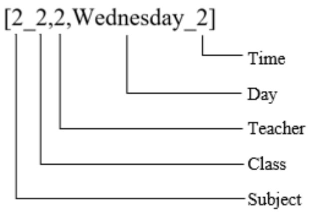


Figure 2. Population Initialization

Individual 1 = ([2_2,2, Wednesday_2], …,[2_3,2,Monday_5])
Individual 2 = ([1_1,2, Thursday_3], …,[3_4,7,Friday_2])
Individual 3 = ([4_6,3, Monday_7], …,[7_1,4,Friday_4])

b. Chromosome evaluation

Chromosome evaluation is a technique for calculating the fitness value which is obtained by adding up the chromosome values in the initial population. The fitness value is the definition of the objective function [12]. Determination of fitness is done by examining the number of collisions that occur when determining the individual produced by each chromosome. The formula for calculating the fitness value is as follows.

$$\text{fitness} = \frac{1}{1+f(x)} \quad (1)$$

Notes:
f(x) = objective function (total weight values of all conflicting values)
The f(x) value used to find the fitness value in this system is as follows.

$$\text{fitness} = \frac{1}{(1+G+K)} \quad (2)$$

Notes:

G = number of clashing teacher chromosomes
K = number of classes clashing chromosomes

The following is the calculation of the fitness value generated by the system.

```
F[0]: 1/(1+31+27) = 0.016949152542373
F[1]: 1/(1+32+34) = 0.014925373134328
F[2]: 1/(1+31+41) = 0.013698630136986
Total F: 0.045573155813688
```

Figure 3. Fitness Value by System

c. Chromosome Selection

Chromosome selection is a technique to get parents from the best chromosomes for further processing [13]. The method used is Roulette Wheel Selection, this method selects each chromosome based on probability. The existing chromosomes contain genes that vary but according to the fitness value. The method then selects chromosomes by generating random values from the distance of all fitness values originating from the population of the test results. The selection process using the Roulette Wheel is carried out in several stages as follows.

1. Calculate the probability value

$$\text{probability} = \frac{Fitness[i]}{Total\_fitness} \quad (3)$$

Notes:

Fitness[i] = Individual fitness value

Total_fitness = Total fitness value

The following image is the probability value generated by the system.

```
P[0] : 0.3719108813018
P[1] : 0.32750361189263
P[2] : 0.30058550680557
```

Figure 4. Proability Value by System

2. Calculating Cumulative Value

In this process calculations are carried out to add up the cumulative results of the probabilities that have been obtained before. How to calculate it is as follows.

$$PK[0] = P[0] \quad (4)$$

$$PK[1] = P[0] + P[1] \quad (5)$$

$$PK[2] = P[0] + P[1] + P[2] \quad (6)$$

Notes:

PK[0] = Cumulative Probability of 1

PK[1] = Cumulative Probability of 2

PK[2] = Cumulative Probability of 3

The following is the result of the cumulative probability value generated by the system.

```
PK[0] : 0.3719108813018
PK[1] : 0.69941449319443
PK[2] : 1
```

Figure 5. Cumulative Probability Value by System

In the picture above, the cumulative probability value for individuals [0] is 0.3719108813018, the cumulative probability for individuals [1] is 0.69941449319443 and the cumulative probability value for individuals[2] is 1. Then the value used on the Roulette Wheel and the number that used is a random number between 0-1.

3. Generate random numbers between 0-1

In this section, random numbers are generated between 0-1 as many as the number of individuals that have been generated before. The following is a random value obtained from the system.

```
R[0] : 0.63688879629452
R[1] : 0.98891713190308
R[2] : 0.22359865914266
```

Figure 6. Random Value by System

The next step is to enter the probability value that was previously obtained into the Roulette Wheel.

Table 1. Value range of chromosome selection

Random Value	Individu	Cumulative Probability
R[2]	PK[0]	0 – 0.3719108813018
R[0]	PK[1]	0.3719108813018 - 0.69941449319443
R[1]	PK[2]	0.69941449319443 - 1

Because the value of R[0] is in the range PK[1], then individual[0] is exchanged for individual[1]. Because the value of R[1] is in the PK[2] range, then individual[1] is exchanged for individual[2]. Because the value of R[2] is in the PK[0] range, then individual[2] is exchanged for individual[0].

d. Crossover

Crossover aims to ensure that the genes shared as well as the genes of the children inherit from the genes of the parents [14]. This step allows for solutions to be created by exchanging genes between individuals and parents that are inherently good and will produce even better offspring.

e. Mutations

Mutations are used within a single individual to swap values of a single gene at random, with a certain probability [15]. In this system, permutations are carried out between each chromosome and the chromosome is selected as 1 individual. With a mutation rate of 0.3

Table 2. Value range of mutation

P	P<Prob Mut	Random 1-327
0.995	N	1
0.563	N	2
0.577	N	3

4. RESULTS AND DISCUSSION

4.1 Individual Value Testing

Individual value testing was carried out to determine differences in the best fitness values based on individual values from 1 to 10 with a total number of generations of 10, crossovers of 0.8 and mutations of 0.3. The results of individual testing can be seen in the following table. Based on Table 3 and Figure 7 it can be concluded that the individual values at values 1 to 10 produce the best fitness value of the third individual.

4.2 Generation Value Testing

Generation value testing was carried out to determine the difference in the best fitness values based on generation values from 1 to 10 with a total of 10 individuals, 0.8 crossover and 0.3 mutations. The results of the generation test can be seen in the following table.

Table 3. Value range of individual testing

No	Number of Resurrected Individuals	Best Fitness Value	Execution Time (Second)	Memory Usage (kb)	Notes
1	1	0.12	3.3527100086212	1755.9296875	fail
2	2	0.2	3.9380328655243	2577.3125	fail
3	3	1	4.5227689743042	2847.515625	succeed
4	4	1	7.4366199970245	5436.921875	succeed
5	5	1	7.8539140224457	5558.1640625	succeed
6	6	1	9.8965239524841	5680.3671875	succeed
7	7	1	12.396386861801	7644.1640625	succeed
8	8	1	12.973363876343	7823.359375	succeed
9	9	1	15.163949012756	7963.390625	succeed
10	10	1	14.723721981049	7940.5625	succeed

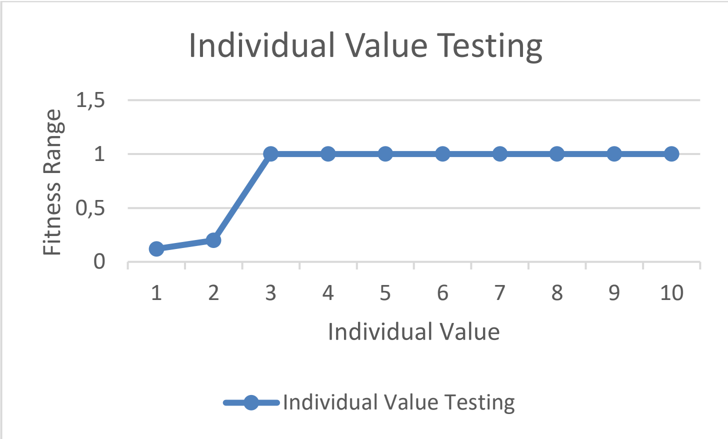


Figure 7. Individual Value Testing Chart

Table 4. Value range of generation testing

No	Number of Resurrected Individuals	Best Fitness Value	Execution Time (Second)	Memory Usage (kb)	Notes
1	1	0.16666666666667	3.7217359542847	1759.9296875	fail
2	2	1	2.9615359306335	1902.984375	succeed
3	3	1	2.984925031662	1702.984375	succeed
4	4	1	2.9312899112701	1770.984375	succeed
5	5	1	2.825933933258	1834.984375	succeed
6	6	1	2.5468208789825	1678.984375	succeed
7	7	1	2.7983589172363	1866.984375	succeed
8	8	1	2.5217869281769	1726.984375	succeed
9	9	1	2.7604060173035	1802.984375	succeed
10	10	1	3.2544419765472	1662.984375	succeed

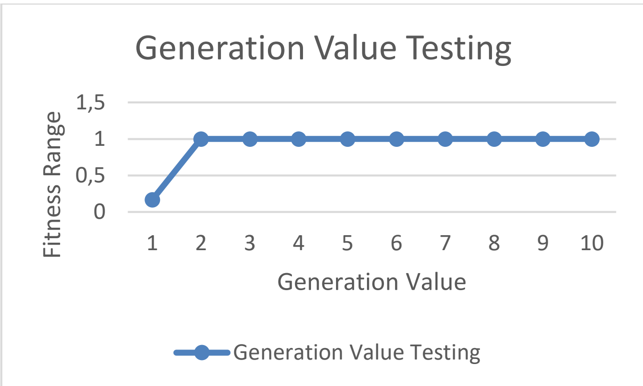


Figure 8. Generation Value Testing Chart

Based on Table 4 and Figure 8 it can be concluded that the generation values at values 1 to 10 produce the best fitness value of the second generation.

4.3 Testing the Crossover Rate Value

Testing the value of the crossover rate is carried out to determine the difference in the best fitness value based on the crossover rate value from 0.1 to 1. The test results can be seen in the following table.

Table 5. Crossover Rate Value Test

No	Number of Resurrected Individuals	Best Fitness Value	Execution Time (Second)	Memory Usage (kb)	Notes
1	0.1	1	13.21267080307	7622.8984375	succeed
2	0.2	1	12.539717912674	7755.7734375	succeed
3	0.3	1	13.692579984665	9775.2421875	succeed
4	0.4	1	14.154518842697	9545.09375	succeed
5	0.5	1	14.115461111069	9731.859375	succeed
6	0.6	1	14.126488924026	9637.453125	succeed
7	0.65	1	17.495476961136	11776.890625	succeed
8	0.7	1	14.417712926865	9674.6328125	succeed
9	0.75	1	15.164895057678	9705.5390625	succeed
10	0.8	1	13.050864934921	9711.328125	succeed
11	0.85	1	12.683351993561	9595.265625	succeed
12	0.9	1	16.556705951691	9715.5390625	succeed
13	0.95	1	17.163483142853	11803.8046875	succeed
14	1	1	15.003257989883	9636.3828125	succeed

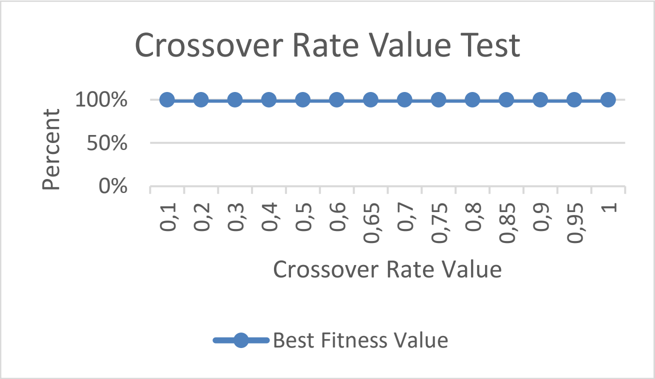


Figure 9. Crossover Rate Value Test Chart

Based on Table 5 and Figure 9 it can be concluded that the value of the crossover rate at values 0.1 to 1 produces the best fitness value because it produces a value of 1.

4.4 Testing the Mutation Rate Value

Testing the value of the crossover rate is carried out to determine the difference in the best fitness value based on the crossover rate value from 0.1 to 1. The test results can be seen in the following table.

Table 6. Mutation Rate Value Test

No	Number of Resurrected Individuals	Best Fitness Value	Execution Time (Second)	Memory Usage (kb)	Notes
1	0.1	1	14.287071943283	9853.0078125	succeed
2	0.2	1	14.85267996788	9689.8671875	succeed
3	0.3	1	11.71119093895	8072.453125	succeed
4	0.4	1	13.291188001633	7828.59375	succeed
5	0.5	1	14.236901044846	10009.3125	succeed
6	0.6	1	12.690559864044	7952.953125	succeed
7	0.65	1	13.599791049957	9715.5078125	succeed
8	0.7	1	12.944355964661	8024.25	succeed
9	0.75	1	14.511309862137	10078.8125	succeed
10	0.8	1	14.685137987137	10049.1796875	succeed
11	0.85	1	14.322882175446	8017.953125	succeed
12	0.9	1	11.178419113159	8147.34375	succeed
13	0.95	1	15.378298044205	10206.4453125	succeed
14	1	1	13.853434801102	10014.390625	succeed

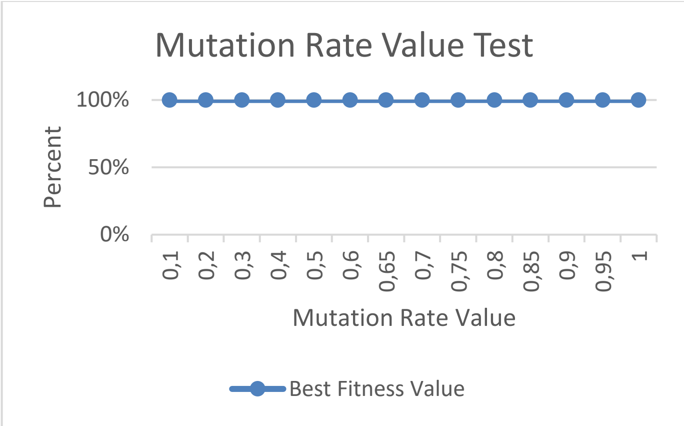


Figure 10. Mutation Rate Value Test Chart

Based on Table 6 and Figure 10 it can be concluded that the value of the mutation rate at values 0.1 to 1 produces the best fitness value because it produces a value of 1.

4.5 Execution Time Testing

Execution time testing is carried out to determine the response to system speed. Response testing is done by receiving and sending data to databases and websites, to get an accurate response is achieved by testing 30 times. The graph of the execution time test results can be seen in the following figure.

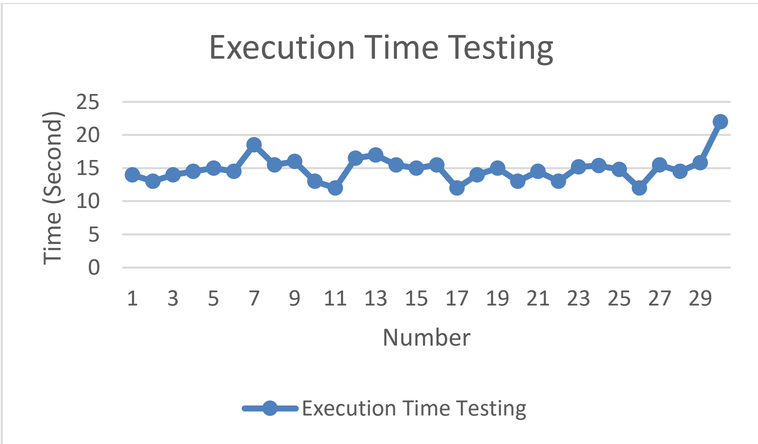


Figure 11. Execution Time Test Chart

The average value obtained from the test is 14.42990657 seconds. The execution time value is obtained from a pre-designed website for each output result which will display the processing time. The calculation of the execution time test can be seen in the attachment.

4.6 Testing The Validity Of The Genetic Algorithm

Testing the validity of the genetic algorithm is done by checking the chromosomes based on the end result of the mutation of the genetic algorithm. In individuals who still have conflicting schedules, chromosomes will appear in bold and red. In individuals who no longer have conflicting schedules, chromosomes will not appear in bold. Apart from looking at the chromosomes, it can also be seen through the fitness value, if the fitness value is 1, it means that the individual results do not have conflicting schedules. If the fitness value is less than 1, it means that the individual results still have conflicting schedules. The examples of schedule chromosomes that do not clash and schedules that clash are as follows.

[1_1,2,Monday_3]	[1_1,2,Tuesday_2]
(a)	(b)

Figure 12. (a) Chromosomes do not clash, (b) Clash chromosomes

After testing 10 times, with values enter the genetic algorithm as follows.

- Number of Individuals = 10
- Number of Generations = 25
- Crossover probability = 0.8
- Mutation Probability = 0.3

The fitness value results are obtained as follows.

Table 7. Genetic Algorithm Validity Test

No	Test	Fitness Value Results	Chromosome Results in Bold	Status
1	First Test	1	Nothing	No Schedule Conflicts
2	Second Test	1	Nothing	No Schedule Conflicts
3	Third Test	1	Nothing	No Schedule Conflicts
4	Fourth Test	1	Nothing	No Schedule Conflicts
5	Fifth Test	1	Nothing	No Schedule Conflicts
6	Sixth Test	1	Nothing	No Schedule Conflicts
7	Seventh Test	1	Nothing	No Schedule Conflicts
8	Eighth Test	1	Nothing	No Schedule Conflicts
9	Ninth Test	1	Nothing	No Schedule Conflicts
10	Tenth Test	1	Nothing	No Schedule Conflicts

Based on Table 7, it shows that in the 10 trials there were no conflicting schedules. By looking at the results of fitness scores and individual checking there were no chromosomes in bold. As for knowing more about the accuracy value, it can be seen in the following calculation.

$$\text{Accuracy} = \frac{\text{Total Success}}{\text{Number of Test Scenarios}} \text{ x } 100\%$$

$$\text{Accuracy} = \frac{10}{10} \text{ x } 100\%$$

$$\text{Accuracy} = 100\%$$

Based on the calculations above, it can be concluded that the accuracy value in testing the validity of the genetic algorithm is 100%.

4.7 Admin User Functionality

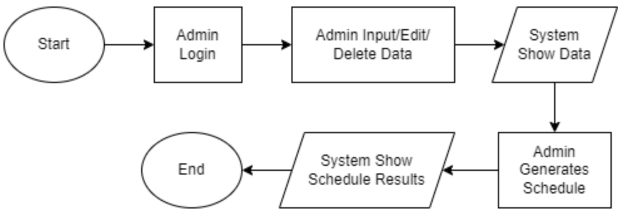
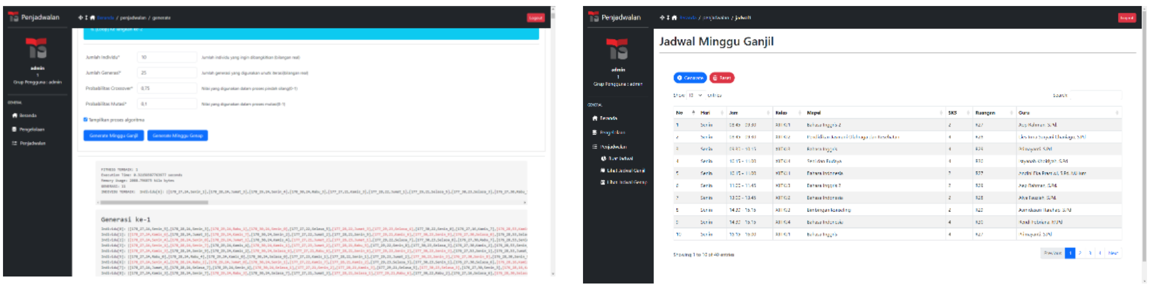


Figure 13. Admin User Functionality

First, the admin user logs in, then it will enter the admin view. In the admin view, the admin can enter, modify, and delete data. The data in question is as follows: subjects, study program, teacher, room, and time. After that the system can display these data and be used for scheduling. Next, the admin generates a schedule, and the system will display the results of the schedule.



(a) (b)

Figure 14. (a) Generate schedule for admin, (b) Admin's schedule

The picture (a) is a schedule generation feature that only the admin user has. In this menu the admin can generate odd and even week schedules. The picture (b) is the result of the schedule for the admin user which is the overall schedule. The schedule is divided into 2, namely the odd week schedule and the even week schedule.

4.8 Teacher User Functionality

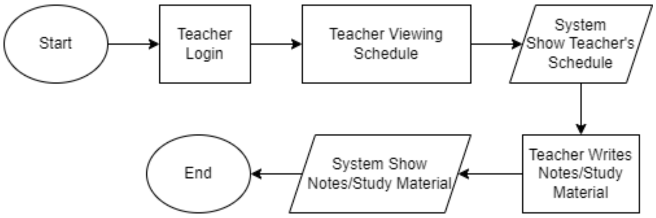
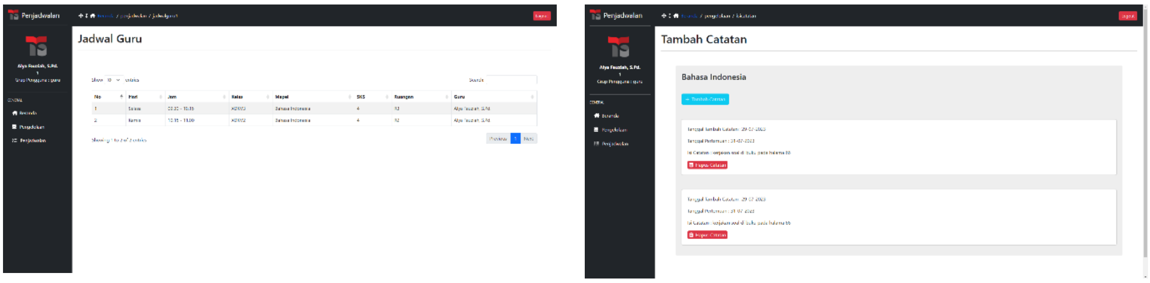


Figure 15. Teacher User Functionality

First the teacher user logs in, then will enter the teacher's view. In the teacher's view, the teacher can see the schedule of the subjects he teaches. Then, the teacher can write notes for students whether class is held or not. In addition, the teacher can enter the material needed for students.



(a) (b)

Figure 16. (a) Teacher's schedule, (b) Teacher's notes

Figure 16a. is the result of the schedule for the teacher user. On the teacher's schedule, only displays the schedule that oversees the teacher concerned. So that each teacher will have a different schedule. Figure 16b. is the notes menu for the teacher user. In the notes menu the teacher can enter notes for students.

4.9 Student User Functionality

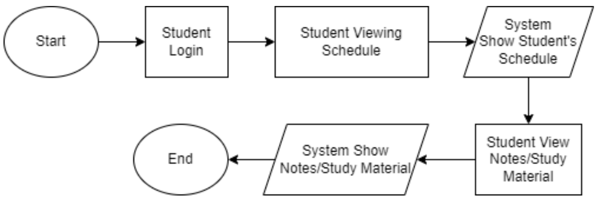
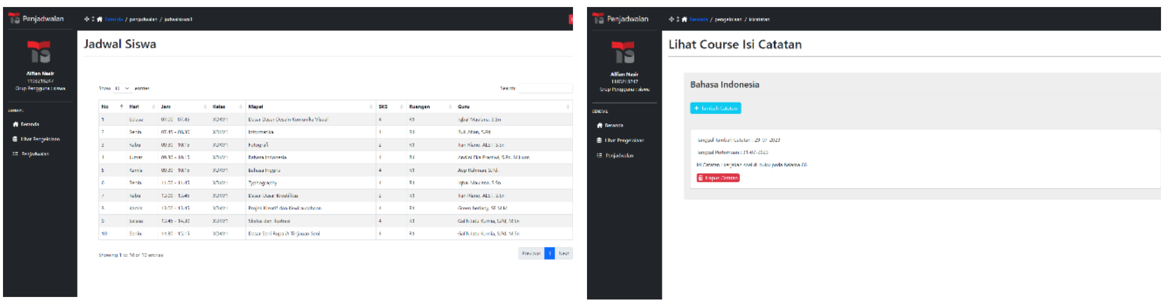


Figure 17. Student User Functionality



(a) (b)

Figure 18. (a) Teacher's schedule, (b) Teacher's notes

First the student user logs in, then will enter the student view. In the student's view, students can see the schedule of their respective subjects. Then, students can see notes from the teacher whether classes are held or not. In addition, students can see the material provided by the teacher.

The picture (e) is the result of the schedule for student users. The schedule displayed is the schedule for each student according to their class. The picture (f) is a note menu for student users. On the note's menu, see notes that have been given by the teacher.

5. CONCLUSION

Based on the results of the research, testing and analysis that has been carried out on the Capstone Project entitled Scheduling Application for Telkom Vocational High School Using Website-Based Genetic Algorithms, it can be concluded that for the genetic algorithm testing, individual values at values 1 to 10 produces the best fitness value for the third individual. The generation value at a value of 1 to 10 produces the best fitness value for the second individual. The crossover rate and mutation rate at a value of 0.1 to 1 produces the best fitness value, which is 1. From 30 execution time testing data, the average value is 14.42990657 seconds and based on 10 genetic algorithm validity testin, the system get 100% accuracy results which means the genetic algorithm can produce an accurate schedule.

REFERENCES

[1] R. A. Aziz, *Pengembangan Sistem Penjadwalan Mata Pelajaran Berbasis Web*, Universitas Negeri Surabaya, 2019.

[2] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran", *J. Media Inform*, vol. 6, no. 1, p. 329, 2022.

[3] H., Rahila, Raghuwanshi, M. dan Jaisawal, N, "Genetic Algorithm Based Clustering," *International Journal of Information System*, 2008.

[4] Anggara, Rahmn. "Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetik: Studi Kasus Program Studi Pendidikan Sekolah Dasar Universitas Pendidikan Indonesia Kampus Bumi Siliwangi". Bandung: Universitas Pendidikan Indonesia, 2012.

[5] Gen, M. and Cheng, R., "Genetic Algorithms and Engineering Optimization", John Wiley & Sons, New York, 2000.

[6] N. I. M. Rojabi, "Analisis Penjadwalan Mata Kuliah dengan Melibatkan Algoritma Genetika", Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2020.

[7] G. S. Budhi, K. Gunadi, and D. A. Wibowo, "Genetic algorithm for scheduling courses", *Commun. Comput. Inf. Sci*., vol. 516, pp. 51–63, 2015.

[8] I. Sbeity, M. Dbouk, H. Kobeissi, "Combining the Analytical Hierarchy Process and The Genetic Algorithm to Solve the Timetable Problem", *International Journal of Software Engineering & Applications*, vol. 5, no. 4, pp 39-50, 2014.

[9] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplkasinya)*, Yogyakarta: Graha Ilmu, ISBN: 979-3289-19-8, 2003.

[10] S. K. A. F. S. N. Sagvan Ali Saleh, "A Genetic Algorithm for Solving an Optimization Problem: Decision Making in Project Management", International Conference on Computer Science and Software Engineering (CSASE), Duhok, Kurdistan Region – Iraq, Duhok, 2020.

[11] R. Paet, "Inisialisasi Populasi pada Algoritma Genetika untuk Travelling Salesman Problem (TSP) Menggunakan Algoritma Steepest Ascent Hill Climbing", Universitas Sumatera Utara, 2020.

[12] F. A. Zainuddin and M. F. A. Samad, "A review of crossover methods and problem representation of genetic algorithm in recent engineering applications," *Int. J. Adv. Sci. Technol*., vol. 29, no. 6, pp. 759–769, 2020.

[13] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations and applications", *Int. J. Eng. Pedagog*., vol. 10, no. 6, pp. 57–77, 2021, doi: 10.3991/IJEP.V10I6.14567.

[14] Z. Michalewicz, Z, *Genetic Algorithme + Data Structure: Evolution Program*, Verlag: Springer, 1996.

[15] R. Murniati, *Penerapam Algoritma Genetika pada DNA Sequencing by Hybridization*. Depok: Departemen Matematika UI, 2009.