IMPLEMENTATION OF GENETIC ALGORITHM IN THE CURRENT SCHEDULING SYSTEM

Pateh Ulum [1*)], Desti Fitriati, [2]

Program Studi Teknik Informatika[1,2]
Fakultas Teknik Universitas Pancasila[1,2]
patchulum@gmail.com [1*], desti.fitriati@univpancasila.ac.id [2]
(*) Corresponding Author

Abstrak
Penjadwalan matakuliah merupakan suatu pekerjaan rutin dalam kegiatan akademik di suatu perguruan tinggi. Dalam pelaksanaannya, proses penjadwalan tidak mudah dilakukan karena banyak faktor yang perlu dipertimbangkan, beberapa faktor yang diperhatikan yaitu seperti kesediaan dosen dalam mengajar, ketersediaan ruang kelas. Selain itu perlu juga diperhatikan jumlah kelas pada setiap matakuliah. Penjadwalan matakuliah merupakan kombinasi antara matakuliah, hari, waktu, ruang kuliah, serta pertimbangan kesediaan waktu dosen untuk mengajar. Untuk mengatasi permasalahan penjadwalan matakuliah dibutuhkan suatu sistem yang dapat menangani proses penjadwalan. Metode yang dapat digunakan untuk menyelesaikan masalah tersebut adalah dengan menggunakan pendekatan Algoritma Genetika. Algoritma genetika merupakan suatu algoritma penjadwalan yang dapat mengkombinasikan waktu dan ruang kuliah secara otomatis dengan menerapkan sistem seleksi alam atau gen. Berdasarkan penelitian yang telah dilakukan, algoritma genetika dapat menyelesaikan masalah penjadwalan dengan cepat, dimana hanya memerlukan waktu 15 detik untuk 78 kelas dan menggunakan sebanyak 16 kromosom. Selain itu, nilai fitness semua kromosom bernilai 0, hal ini menyatakan bahwa hasil penjadwalan yang diperoleh bernilai optimal.

Kata kunci: Sistem Penjadwalan, Penjadwalan Mata Kuliah, Algoritma Genetika

Abstract
Scheduling courses is a routine job in academic activities at a college. In its implementation, the scheduling process is not easy to do because many factors need to be considered, several factors that are considered, such as the willingness of lecturers to teach, the availability of classrooms. Besides that, it is also necessary to pay attention to the number of classes in each subject. Course scheduling is a combination of courses, days, time, lecture space, and consideration of lecturers' willingness to teach. To solve the course scheduling problem, a system that can handle the scheduling process is needed. The method that can be used to solve this problem is to use the Genetic Algorithm approach. The genetic algorithm is a scheduling algorithm that can combine lecture time and space automatically by applying a natural or gene selection system. Based on the research that has been done, the genetic algorithm can solve scheduling problems quickly, which only takes 15 seconds for 78 classes and uses as many as 16 chromosomes. Also, the fitness value of all chromosomes is 0, this means that the scheduling results obtained are optimal.

Keywords: Scheduling System, Course Scheduling, Genetic Algorithms

INTRODUCTION

The course scheduling activity is an activity that is very important for the implementation of a good and orderly teaching and learning process (Suhainingsih, 2015) Where the learning process involves facilities and resources, not only involving lecturers who teach, but also students who will learn. In the process of making a lecture schedule, several things are quite complicated, such as allocating courses, class schedules related to the day and time, space or place of the lecture, and also the lecturer who will teach the course. Where in the allocation process, it must ensure that there are no conflicts between courses, lecture time, and lecture halls (Sari, Alkaff, Wijaya, Soraya, & Kartikasari, 2019).

The process of making lecture schedules at the Faculty of Engineering, Pancasila University is currently managed by the faculty academic staff. Where in the process of making a lecture schedule through several steps, first the study program staff will make a lecture schedule that contains courses and also teaching lecturers, then the data will be submitted to the faculty academic staff. Then after that, the academic staff will summarize each schedule given by the study program and managed using the Microsoft Office Excel application (Azizah & Ramadhani, 2011). The process of making this lecture schedule will take a very long time (R. F. Nugraha, 2018), and one must also be careful in paying attention to the lecture hall used and the time set (Salim, 2016).

To facilitate the activities of making and preparing lecture schedules, an information system that can solve scheduling problems can be made and can arrange class schedules automatically according to predetermined limits (Mittal, Doshi, Sunasra, & Nagpure, 2015). With this course scheduling information system, the process of preparing lecture schedules will be very easy and fast where academic staff only need to enter course data, lecturer data, lecture room data, lecture classes, and request for lecturer schedules if any.

To create a scheduling system automatically, an algorithm is needed that can solve scheduling problems. One of the algorithms that can be used is the genetic algorithm (Laksono, Catur Utami, & Sugiarti, 2016) .With genetic algorithms, the scheduling process no longer takes a long time like making class schedules manually. With this genetic algorithm, class schedules which are a combination of courses, lecturers who teach, lecture halls, and lecture times and days will be randomized and a combination made according to the limitations that have been determined at the time of making the system.

Genetic algorithms are algorithms that attempt to apply an understanding of natural evolution to problem-solving tasks (Setiawan, Herwindiati, & Sutrisno, 2019). The approach taken by this algorithm is to randomly combine various optimal solution choices in a collection to get the next best solution generation, namely in a condition that maximizes its suitability and conforms to predetermined limits (D. W. Nugraha, E.Dodu, & Saud, 2017).

In this study, the lecture scheduling system is made in the form of a desktop application, where the programming language that will be used is the Java programming language and uses the Apache Netbeans IDE 11.3 software as a tool for writing lines of code, besides that MySQL is also used as database processing.

RESEARCH METHODS

Types of Research

This research is experimental research, where the results obtained are based on the best test results.

Time and Place of Research

This research was conducted in July 2020, which is located at the Faculty of Engineering, Pancasila University.

Procedure

The method used in this research is the genetic algorithm method. First, the data needs to be represented so that the data can be understood by genetic algorithms. The process of representing data is called encoding. After the data is represented, the data will be obtained by the genetic algorithm, before explaining the stages of the genetic algorithm, several terms in the genetic algorithm must be known, the following are explained in Table 1:

Table 1. Genetic Algorithm Terms

No.	Keywords	Definition
1.	Genes	Genes represent the day, time, and space specified for lecture classes
2.	Allele	Represents class Subject
3.	Chromosomes	Collection of Genes, in other words, chromosomes is a course schedule for one week
4.	Population	A collection of course schedules

After knowing the terms in the genetic algorithm, the following are several processing stages, namely building the initial population, evaluating the value of fitness, selection, crossbreeding, and mutation. The following is an explanation of each process in the genetic algorithm:

1) Initial Population

Course scheduling includes components of courses, lecturers, days, hours (time), and lecture halls. To make chromosome models there are several components, namely subjects, days, hours, and space. The initial population formation of course scheduling is n chromosomes. Where n is a candidate for the number of courses. Each gene on a chromosome represents one lecture.

2) Evaluate the value of Fitness

After a chromosome is formed, the next step is to determine the Fitness Value for each chromosome. The fitness value is a determinant of whether the chromosome is good or not. The Fitness value is obtained from the number of Hard Constraint violations committed on 1 chromosome. Figure 1 below is the evaluation flow of Fitness Values.

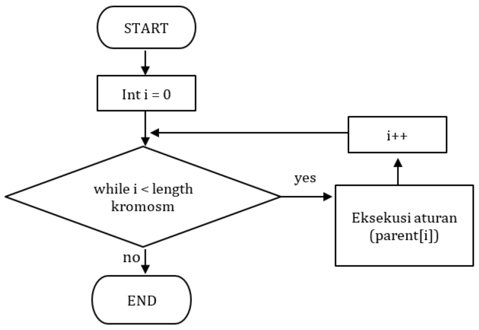


Figure 1. Fitness Value Evaluation Flow

The rules (constraints) and Fitness weights that are applied in this scheduling system are as follows:

Table 2. Rules in the scheduling system

HC	Rules	Fitness Value
1	One room can only be used by one course on certain days and hours	Offense = 1 if there are two courses in the same room, hour and day
2	A lecturer can only teach one subject at the same time	Violation = 1 if there are two or more courses taught by one lecturer
3	Subjects for a certain semester cannot be at the same time except for the same subject and different lecturers	Violation = 1 if there are two or more same courses for the same lecturer at the same time

In the evaluation stage of Fitness values, chromosomes can be said to be optimal or a perfect solution if the Fitness value is equal to 0 (zero). The flowchart will start by executing all chromosomes. Each chromosome will be executed to find the Fitness value by running the Rule Executionmethod().

3) Selection
Selection is a process that will determine the components that will become candidate parents to form a new chromosome. In this selection process, a selection model is carried out using the Roulette Wheel Selection method concerning Fitness. The smaller the Fitness value, the more chromosomes are selected as parents for the next population. Figure 2 below is the flow of the algorithm for sorting chromosomes based on the high Fitness value:

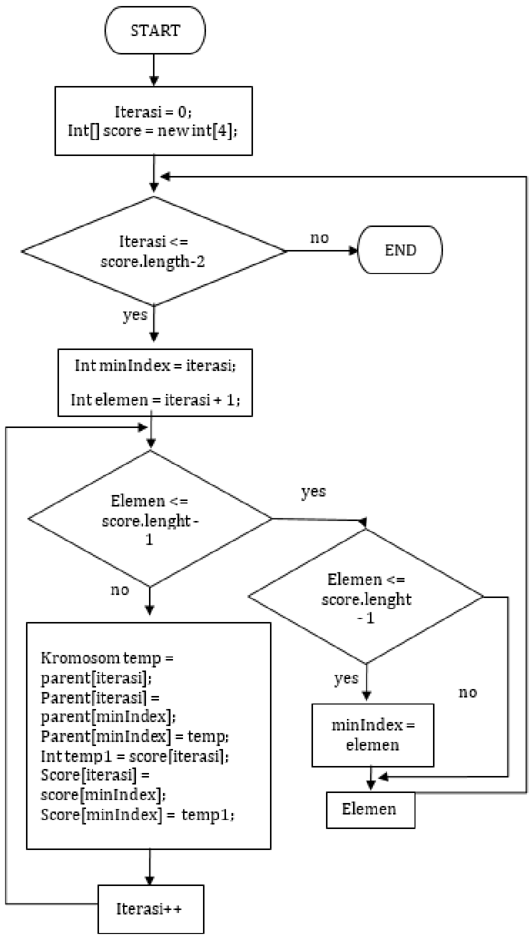

START
Iterasi = 0;
Int[] score = new int[4];
Iterasi <= score.length-2
no
END
yes
Int minIndex = iterasi;
Int elemen = iterasi + 1;
Elemen <= score.lenght - 1
yes
no
Elemen <= score.lenght - 1
yes
no
Kromosom temp = parent[iterasi];
Parent[iterasi] = parent[minIndex];
Parent[minIndex] = temp;
Int temp1 = score[iterasi];
Score[iterasi] = score[minIndex];
Score[minIndex] = temp1;
minIndex = elemen
Elemen
Iterasi++

Figure 2. Selection of Chromosomes with the Highest Fitness Value

4) Crossover
Crossover is the process of combining two chromosomes from two selected parents. The result of this combination will later form new chromosomes. The system is made using 4 chromosomes in 1 population where the 1st and 2nd chromosomes are slots that are reserved for the selected parent candidate. Chromosome selection is carried out based on Roulette While Selection. Meanwhile, the 3rd and 4th chromosomes are the chromosome slots that will be used for child candidate slots. Child chromosomes (3 and 4) in the previous iteration will be replaced by new chromosomes in the next iteration. This process will occur based on natural selection. For the scheduling case of this course, the One Point Crossover method is applied. Figure 3 below is the crossover algorithm applied in this study.

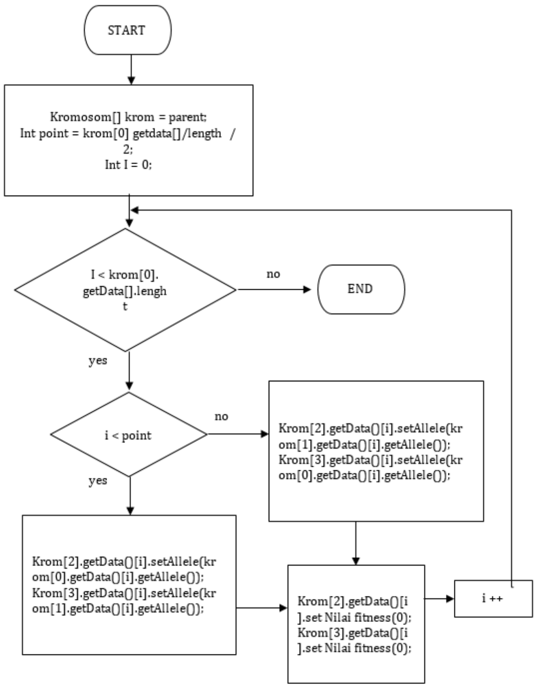
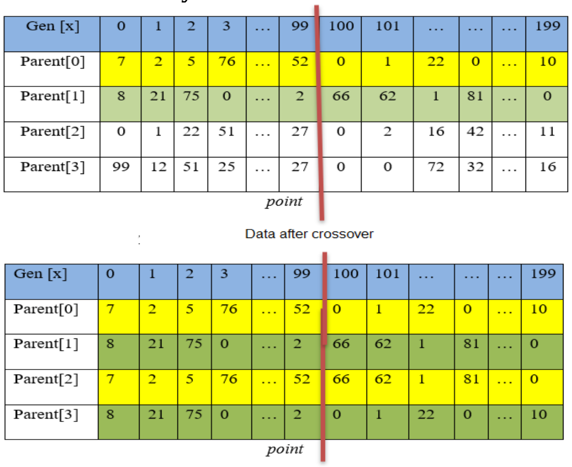

START
Kromosom[] krom = parent;
Int point = krom[0] getdata[]/length /
2;
Int I = 0;
I < krom[0].
getData[].lengh
t
no
END
yes
i < point
no
Krom[2].getData()[i].setAllele(krom[1].getData()[i].getAllele());
Krom[3].getData()[i].setAllele(krom[0].getData()[i].getAllele());
yes
Krom[2].getData()[i].setAllele(krom[0].getData()[i].getAllele());
Krom[3].getData()[i].setAllele(krom[1].getData()[i].getAllele());
Krom[2].getData()[i].set Nilai fitness(0);
Krom[3].getData()[i].set Nilai fitness(0);
i ++

Figure 3. Crossover Algorithm

Figure 3 above explains that each child chromosome (chromosome 2 and 3) is a combination of genes from the two-parent chromosomes (chromosome 0 and chromosome 1). In the One Point Crossover method, the determining point of descent is the point where the point value is half the length of each chromosome. In this scheduling system, the nGen on each chromosome is 200. So that the points for each chromosome are:

Point = nGen / 2
= 200 / 2
= 100

Figure 4 below is an illustration of the crossover used in this study.



Gen [x]	0	1	2	3	...	99	100	101	...	...	...	199
Parent[0]	7	2	5	76	...	52	0	1	22	0	...	10
Parent[1]	8	21	75	0	...	2	66	62	1	81	...	0
Parent[2]	0	1	22	51	...	27	0	2	16	42	...	11
Parent[3]	99	12	51	25	...	27	0	0	72	32	...	16

point

Data after crossover

Gen [x]	0	1	2	3	...	99	100	101	...	...	...	199
Parent[0]	7	2	5	76	...	52	0	1	22	0	...	10
Parent[1]	8	21	75	0	...	2	66	62	1	81	...	0
Parent[2]	7	2	5	76	...	52	66	62	1	81	...	0
Parent[3]	8	21	75	0	...	2	0	1	22	0	...	10

point

Figure 4. Crossover Illustration

5) Mutation

The mutation process is the exchange of genes from one gene to another in a chromosome. The mutations used in the scheduling system are ordinary. Ordinary mutations are mutations that do not pay attention to the Fitness value on a chromosome. All genes have an equal probability of being mutated.

Data, Instruments, and Data Collection Techniques

The data used is in the form of secondary data (pre-existing data) and is managed by the academic staff of the Faculty of Engineering, Pancasila University. The data used are in the form of lecturer data, classroom data, course data, and data on the number of credits for each course. The data was obtained from the Faculty of Engineering academic. These data are used in the creation of a course scheduling system using Genetic Algorithms.

Data analysis technique

In this study, evaluation is carried out by testing with the Black box testing method, this method applies testing by running or executing modules on the system being made, then the module is observed whether it is by the desired business process or not. The expected result of this scheduling system is an optimal lecture schedule (has predefined rules), the optimal scheduling has a value of Fitness = 0 on each chromosome.

RESEARCH RESULTS AND DISCUSSION

Application Scenarios The method used

The initial population is generated randomly and then the evaluation process is carried out for each chromosome. The length of one chromosome is n genes. where n is the product of the number of lecture days, the number of rooms available, and the amount of time available. Mapping on chromosomes will apply the time slot in the course hour, each time slot consists of 2.5 hours, assuming each subject has 3 credits (1 credit = 50 minutes, so 3 credits = 3 * 50 minutes = 2.5 hours). Table 2 below is a grouping of hours in one day.

Table 3. Schedule Grouping

Code	Time
1	08.00 – 10.29
2	10.30 – 12.59
3	13.00 – 15.29
4	15.30 – 17.59

The table provides a brief description of time grouping within one day. If 10 hours are used for

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v3i2.57

each room, then the mapping of the space in the chromosomes is 4 times for each room in 1 day. Each time slot will be presented with 1 gene. Where genes are candidates for a particular lecture class. Mapping to determine the length of a chromosome is by counting the number of lecture days (h) * the number of rooms (r) * the number of lectures for 1 day in a room (w). If the formula is formed, we get ngen = h * r * w. For example, in the Engineering faculty, there are 5 lecture days with 10 lecture halls and in one day there are 4 lectures in one room, the chromosome length is ngen = 5 * 10 * 4, which is 200 genes.

In system design, n-gen is a representation of the length of 1 chromosome, where the chromosome is a 1-week course schedule. Apart from the lecture time component, class lectures are also an important component in the scheduling system. Where the lecture class table consists of the class code, course code, lecturer code, and also the class. The college class data will be implemented into chromosomes, each class will be assigned a random gene index. This means that each class has a different gene index (slot). Where the gene index is not occupied by any college class will be coded with 0, which means that there is no class at that time. Table 3 below is a mapping of a chromosome that is formed.

Table 4. Chromosome Mapping

GenX	0	1	2	3	4	5	……..	200
Allele	1	5	7	0	1	24	……..	56

In Table 3 above, it can be seen that genes represent the entire time slot, while allele is the value that contains the lecture class. The following is the overall chromosome mapping by combining the day, space, time, and class of lectures within one lecture week:

Gen [0]: Class 2 (For example, Mathematics code Alg1 - Algorithm and programming subject 1, semester 3 Class A lecturer Sri Rezeki CN) Monday, Room 1 (for example room 211A), Time slot 1 (time 08.00 - 10.29 wib)

Gen [33]: Class 30 (For example, Mathematics code Alg2 - Algorithm and Programming course 2, semester 5 Class G Gregoriu teaching lecturer) Monday, room 10, time slot 3 (14.00 - 16.59 WIB)

After getting the chromosomes, the chromosomes will then be inserted into the population, in other words, the population is a collection of course schedules. In program design, 1 population consists of 4 chromosomes. If it is illustrated, the population design built for iteration 1 (initial population) is as shown in Figure 4 below.

Iterasi 1														
Gen[x]	0	1	2	3	4	5	6	7	8	9	10	11	…	199
K1	2	0	0	5	0	0	0	7	0	1	0	3	…	6
K2	0	0	1	64	0	0	10	2	3	7	44	0	…	0
K3	2	9	0	5	0	1	0	22	0	8	0	0	…	7
K4	100	2	24	0	6	0	120	0	8	0	0	99	…	76

Figure 5. Initial Population Development

Building the initial population also considers the soft constraints that apply. At the time of building the initial population, the demand lecture classes will be pre-entered into the system according to the Day, Space, and Time by the request. Then the other lecture classes will be entered randomly. The following is the Random Process algorithm:

Random Algorithm

```
1.  Create Variable r of type random
2.  Create a Variable day of type int. Set value as 5
3.  Create a time variable of type int. Set value as 4
4.  Create a matkul variable of type int set with the length
    of the lecture class
5.  Create a gene named data throughout (day * space *
    time)
6.  Create a Boolean variable called to check which
    functions to check whether the college class is on the
    chromosome or not
7.  Create a variable N of type int set with matkul
8.  For I = 0 to data length, do step 9
9.  Create a new blank gene set equal to blank courses and
    space-time days according to the gene location
10. For i = 0 until matkul do step 11
11. Set all checks to false. This indicates that the course is
    not yet on the chromosome.
12. For i = 0 to the length of the request data, perform steps
    13-15
13. Put the whole class of lecture into the gene according to
    the request Day, Space and time.
14. Set class check requests with true. Indicates that the
    course has been in the chromosome
15. Subtract N as many as the Requests course
16. Create an index variable of type int set with 0
17. When N does not equal 0. Perform steps 18-22
18. Create a capture variable of type int set with a random
    value of data length.
19. Checks whether the check to index is true. If yes, then
    increase the value by 1. Otherwise, go to steps 20-22
20. Check whether the catch position gene slot is empty. If
    so, enter the course class under Gen.
21. Add index value with 1.
22. Subtract 1 from the value of N
23. Return the data as a return value.
24. 24. Done
```

After that, continue by evaluating the fitness value by running the RuleExecution() method. This method is a representation of the Hard Constraint rule. Here is the RuleExecution() method:

```
RuleExecution Algorithm
1. Create a variable named parent of type chromosome.
2. Call the CekDosen method with the parameter x. the results are stored in the parent.
3. Call the checkMatakuliah method with the parent parameter. The results are stored in the parent.
4. Return the parent value;
5. Done
```

In the RuleExecution () method, the chromosomes will be evaluated for each gene. 3 methods present the course scheduling rules that apply to this research, namely:

1. Hard Constrans (HC) 2 method CekDosen (parameter x is chromosome type)

```
Algorithm Hard Constrans (HC) 2 method CekDosen
1. Create a variable position_hours of type array of int with the length corresponding to the number of lecture spaces.
2. Create a day variable of type array of string with input Monday to Friday.
3. Create a clock variable of type int with an initial value of 1.
4. Create a count variable of type int with an initial value of 0.
5. For i equal to 0 to the length of the hour position, perform step 6.
6. If I is equal to 0, then set hour_ position to i is equal to 0. Also, set hour_ position i is worth hour_ position to [i - 1] plus 4.
7. When true, do step 8.
8. If the Gen to count on the x chromosome is the same as the day 0 (Monday), then add the count value to 1. Also, stop looping.
9. For i = 1 to the length of the day, do step 10-22.
10. For b = 0 to less than 4 perform steps 11-19.
11. For j = 0 to less than the amount of space, perform steps 12-17.
12. For k = j + 1 up to the number of spaces, perform steps 13-17.
13. Create a variable of 1 and 2 of type String. Where appeal 1 store the Lecturer id from gene to the j_hours_ position and appeal 2 stores the Lecturer id from gene to the hour_ position to k.
14. Check if the appeal of 1 is equal to "-". if it is then the program does nothing. If not, go to step 15.
15. Check if appeal 1 is the same as appeal 2. If yes then go through steps 16-17.
16. Create a Fitness_ variable with type int which contains the Fitness value from the gene data to the hour_ position to k.
17. Set gene Fitness value to hour_ position to k with Initial Fitness +1
18. For 1 = 0 to hour_ position, perform step 19.
19. Set hour_ position to 1 equal to hour_ position to 1 + 1.
20. For 1 = 0 to the hour_ position length, perform step 21.
21. Check if 1 is equal to 0. If yes, then 1 hour_ position = count. If not then hour_ position 1 is equal to hour_ position 1-1 plus 4.
22. Set count equal to count + count.
23. Return the x value.
24. Done
```

2. Hard Constrans (HC) 3 method CekMatakuliah (parameter x is chromosome type)

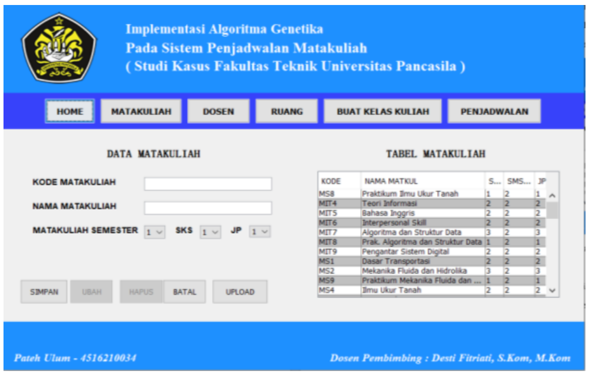
```
Algorithm Hard Constrans (HC) 3 method CekMatakuliah
1. Create a variable array of int hours_position along the space_count.
2. Create a variable for the day of type array of String with input is College day (Monday-Friday).
3. Create a clock variable of type int set with value 1.
4. Create variable count of type int set with 0;
5. For i equal to 0 to the length of the hour_ position do step 6.
6. Check if i is equal to 0, if yes then set hour_ position to i is equal to 0. If not then set hour_ position to i is equal to hour_ position i - 1 plus 4.
7. When true, do step 8.
8. Check whether the day of the x chromosome gene count is the same as day 0 (Monday). If yes, then the count is added by 1. If not then stop the loop.
9. For i = 1 to the length of the day perform steps 10-27.
10. For b = 0 to 4 do steps 11-24.
11. For j = 0 to a number of spaces, perform steps 12—21.
12. For k = j + 1 to sum_space, perform steps 13-21.
13. Create a variable of 1 of type String set with chromosome_id_id.x gen to j_hour_ position.
14. Create a variable of 2 of type String set with chromosome_id_id.x gen to hour_ position to k.
15. Create a variable of 3 of type String set with chromosome semester x gen to j_hour position.
16. Create a variable of 4 of type String set with chromosome semester x gen to hour_ position to k.
17. Create a variable of 7 of type String set with chromosome class x gen to j_hour_ position.
18. Create a variable of 8 of type String set with chromosome class x gen to hour_ position to k.
19. Check if the appeal of 1 is equal to "-". if yes then do nothing. If not, do step 20 - 21.
20. Check whether 3 is equal to 4, if yes, check whether 7 is equal to 8, if yes then set the initial Fitness variable with the fitness value of chromosome x gen to hour_ position to k.
21. Set the x chromosome gene data to hour_ position to k with the value Fitness + 1
22. For i = 0 to the 1st-hour_ position, perform step 23.
23. Check if i is equal to 0. If yes, then 1 hour_ position = count. If not, then the 1-hour_ position equals the i-1 hour_ position plus 4.
24. Set count equal to count + count.
25. For i = 0 to hour_ position, do step 26.
26. Check if i is equal to 0, if yes then i_hour_ position equals count. If not, then i clock_ position is the same as i + i hour_ position plus 4.
27. Set count equal to count + count.
28. Return the x value
29. Done.
```

After evaluating the value of Fitness, a regeneration process will be carried out in the scheduling system. In the regeneration process, the existing genes change to the next generation. This regeneration process includes the process of selection, crossover, and mutation.

System Implementation

The main features in the system include processing data for courses, lecturers, classrooms, and making class schedules. Here are some screenshots of the features available.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v3i2.57


Implementasi Algoritma Genetika
Pada Sistem Penjadwalan Matakuliah
(Studi Kasus Fakultas Teknik Universitas Pancasila)

DATA MATAKULIAH

KODE MATAKULIAH
NAMA MATAKULIAH
MATAKULIAH SEMESTER 1 SKS 1 JP 1

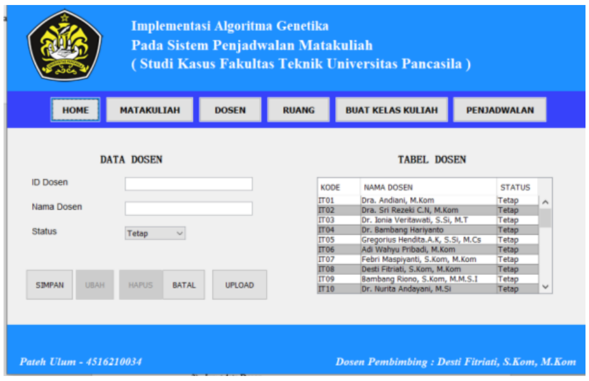
TABEL MATAKULIAH

KODE	NAMA MATKUL	S...	SMS...	JP
MS8	Praktikum Ilmu Ukur Tanah	1	2	1
MIT4	Teori Informasi	2	2	2
MIT5	Bahasa Inggris	2	2	2
MIT6	Interpersonal Skill	2	2	2
MIT7	Algoritma dan Struktur Data	3	2	3
MIT8	Prak. Algoritma dan Struktur Data	1	2	1
MIT9	Pengantar Sistem Digital	2	2	2
MS1	Dasar Transportasi	2	2	2
MS2	Mekanika Fluida dan Hidrolika	3	2	3
MS9	Praktikum Mekanika Fluida dan ...	1	2	1
MS4	Ilmu Ukur Tanah	2	2	2

Pateh Ulum - 4516210034   Dosen Pembimbing : Desti Fitriati, S.Kom, M.Kom

Figure 6. Course Data Management Feature

Figure 6 above is a master feature of the course, where the process of adding, editing, and deleting is available there. Also, the upload button is an option that users can use if they have data in excel format without having to add it one by one. Likewise for the lecturer data feature as seen in Figure 7 below.


Implementasi Algoritma Genetika
Pada Sistem Penjadwalan Matakuliah
(Studi Kasus Fakultas Teknik Universitas Pancasila)

DATA DOSEN

ID Dosen
Nama Dosen
Status Tetap

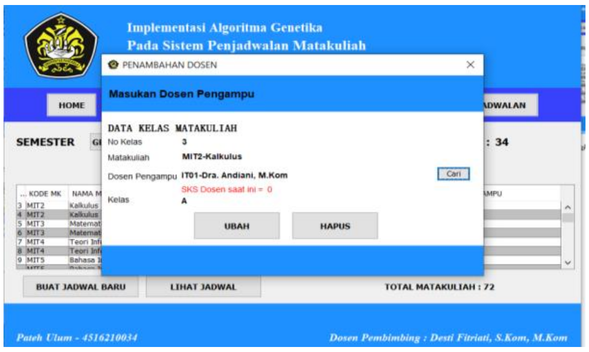
TABEL DOSEN

KODE	NAMA DOSEN	STATUS
IT01	Dra. Andiani, M.Kom	Tetap
IT02	Dra. Sri Rezeki C.N, M.Kom	Tetap
IT03	Dr. Ionia Veritawati, S.Si, M.T	Tetap
IT04	Dr. Bambang Hariyanto	Tetap
IT05	Gregorius Hendita.A.K, S.Si, M.Cs	Tetap
IT06	Adi Wahyu Pribadi, M.Kom	Tetap
IT07	Febri Maspiyanti, S.Kom, M.Kom	Tetap
IT08	Desti Fitriati, S.Kom, M.Kom	Tetap
IT09	Bambang Riono, S.Kom, M.M.S.I	Tetap
IT10	Dr. Nurita Andayani, M.Si	Tetap

Pateh Ulum - 4516210034   Dosen Pembimbing : Desti Fitriati, S.Kom, M.Kom

Figure 7. Lecturer Data Management Feature

After the master data is entered into the system, then the next step is to build a class where the process is carried out by combining the lecturer with the courses being taught. Also, this feature places optional restrictions, such as some lecturers who can only teach on certain days and so on. Figure 8 below shows the process carried out in processing a lecture class.


Implementasi Algoritma Genetika
Pada Sistem Penjadwalan Matakuliah

PENAMBAHAN DOSEN

Masukan Dosen Pengampu

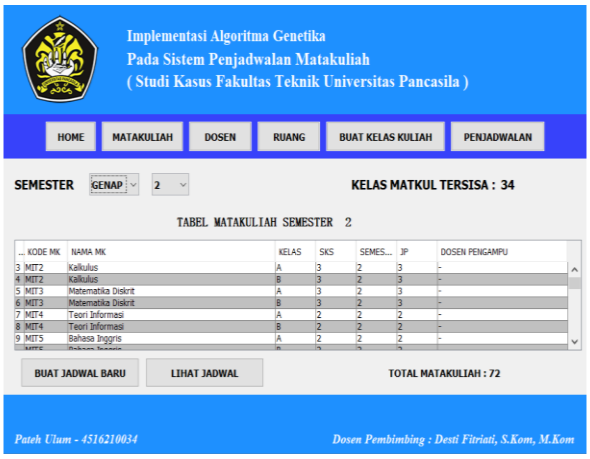
DATA KELAS MATAKULIAH
No Kelas 3
Matakuliah MIT2-Kalkulus
Dosen Pengampu IT01-Dra. Andiani, M.Kom
SKS Dosen saat ini = 0
Kelas A

SEMESTER
: 34
TOTAL MATAKULIAH : 72

Pateh Ulum - 4516210034   Dosen Pembimbing : Desti Fitriati, S.Kom, M.Kom

Figure 8. Features of Manage Class Classes

After all the required data (such as courses, lecturers, rooms, class lectures) have been stored in the system, then an automatic schedule is made using a genetic algorithm by selecting the "scheduling" button. The result of this feature is a class schedule that is used as a user recommendation in making schedules. Figure 9 shows the class schedule obtained from the system trial.


Implementasi Algoritma Genetika
Pada Sistem Penjadwalan Matakuliah
(Studi Kasus Fakultas Teknik Universitas Pancasila)

SEMESTER GENAP 2   KELAS MATKUL TERSISA : 34

TABEL MATAKULIAH SEMESTER 2

...	KODE MK	NAMA MK	KELAS	SKS	SEMES...	JP	DOSEN PENGAMPU
3	MIT2	Kalkulus	A	3	2	3	-
4	MIT2	Kalkulus	B	3	2	3	-
5	MIT3	Matematika Diskrit	A	3	2	3	-
6	MIT3	Matematika Diskrit	B	3	2	3	-
7	MIT4	Teori Informasi	A	2	2	2	-
8	MIT4	Teori Informasi	B	2	2	2	-
9	MIT5	Bahasa Inggris	A	2	2	2	-

TOTAL MATAKULIAH : 72

Pateh Ulum - 4516210034   Dosen Pembimbing : Desti Fitriati, S.Kom, M.Kom

Figure 9. Scheduling System Results

Results Evaluation

Based on the results of the research, it can be concluded that the genetic algorithm can be used to solve scheduling problems automatically, this algorithm can overcome quite complicated scheduling with the limitations of predetermined rules. Schedule time is also shorter, reaching only 15 seconds for 78 classes. Of course, this is shorter when compared to conventional methods which can take more than 7 days. Also, this study obtained results where the fitness value for all chromosomes is 0. This proves that this study produces optimal scheduling because it is indicated by the fitness value for each chromosome is 0. Figure 10 below is a capture of the layer from the scheduling process time.

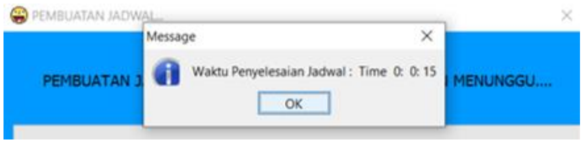

PEMBUATAN JADWAL
Message
Waktu Penyelesaian Jadwal : Time 0: 0: 15

Figure 10. Scheduling Processing Time

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

This research was conducted to solve problems that arise in scheduling lectures, where there are many limitations such as time available for some lecturers who can only teach on certain days, the number of credits that must be covered by each lecturer, and so on. This resulted in the creation of a schedule that was more complex and time-consuming. Therefore, the solution is given in this study which builds an automatic scheduling system based on available constraints. The results showed that the time generated to make a schedule was only 15 seconds for the case study of 78 classes. In addition, the fitness value for the 16 chromosomes is 0, this proves that according to the GA algorithm, the fitness value of all chromosomes that is 0 is stated to have optimal scheduling.

Recommendations

With the scheduling system, scheduling will be easier, however, it should be noted that the rules in the scheduling system need to be defined to be applied to the system being built. For this reason, it is necessary to add class capacity rules to the scheduling system, to regulate class capacity that can be used for certain classes, this takes into account the room capacity which is not always the same in a university or institution. Also, it is necessary to set a time limit by the number of credits in each course so that there is no waiting time between courses when the course has a short time.

REFERENCE

Azizah, N., & Ramadhani, Y. (2011). Pembangunan Sistem Informasi Penerimaan Siswa Baru Di Sekolah Menengah Kejuruan Al-Irsyad Tegal. *Journal Speed – Sentra Penelitian Engineering Dan Edukasi*, *3*(3), 35–43. Retrieved from http://www.ijns.org/journal/index.php/speed/article/view/1258

Laksono, A. T., Catur Utami, M., & Sugiarti, Y. (2016). Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran Dan Kesehatan Universitas Muhammadiyah Jakarta). *Studia Informatika: Jurnal Sistem Informasi*, *9*(2), 188. https://doi.org/10.15408/SIJSI.V9I2.7647

Mittal, D., Doshi, H., Sunasra, M., & Nagpure, R. (2015). Automatic Timetable Generation using Genetic Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(2), 245–248. Retrieved from https://ijarcce.com/wp-content/uploads/2015/03/IJARCCE4I.pdf

Nugraha, D. W., E.Dodu, A. . Y. ., & Saud, A. T. . . (2017). Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus Pada Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako). *JIMT*, *14*(2), 242–255. Retrieved from https://bestjournal.untad.ac.id/index.php/JIMT/article/view/9026/

Nugraha, R. F. (2018). *Sistem Penjadwalan Otomatis Kuliah Mahasiswa* (Universitas Muhammadiyah Surakarta). Universitas Muhammadiyah Surakarta. Retrieved from http://eprints.ums.ac.id/64403/

Salim, A. (2016). *Aplikasi Jadwal Mata Kuliah Teknik Informatika dan Sistem Informasi dan Penjadwalan Ruangan Kuliah Berbasis Dekstop* (Universitas Islam Negeri Alauddin Makassar). Universitas Islam Negeri Alauddin Makassar. Retrieved from http://repositori.uin-alauddin.ac.id/id/eprint/2298

Sari, Y., Alkaff, M., Wijaya, E. S., Soraya, S., & Kartikasari, D. P. (2019). Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik Tournament Selection. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *6*(1), 85–92. https://doi.org/10.25126/jtiik.201961262

Setiawan, J. Y., Herwindiati, D. E., & Sutrisno, T. (2019). Algoritma Genetika Dengan Roulette Wheel Selection dan Arithmetic Crossover Untuk Pengelompokan. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKSI)*, *7*(1), 58–64. Retrieved from https://journal.untar.ac.id/index.php/jiksi/article/view/5882

Suhainingsih, D. M. (2015). *Analisa Pengembangan Online Jurusan Rencana Studi Plus (OJRS+) Sebagai Media Batal Tambah Penjadwalan Mahasiswa Studi Kasus: Perguruan Tinggi Raharja*. Tangerang. Retrieved from https://widuri.raharja.info/index.php?title=Backup_Maya