Scheduling System for Proposal Seminars and Final Project Examination Using Genetic Algorithms for Final Project Information Systems

Muhammad Burhanudin[1], Ginanjar Suwasono Adi[2*], Putri Elfa Mas'udia[3]

[1,2,3]Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1]burhanudinm81@gmail.com , [2]ginanjar.adi@polinema.ac.id, [3]putri.elfa@polinema.ac.id

Abstract— **The process of scheduling seminar proposals and final project examination in academic environments often faces highly complex obstacles, such as lecturer schedule conflicts, room limitations, and difficulties in matching the expertise of examiners with the topics of students' final projects. This study aims to develop a web-based automatic scheduling system using genetic algorithms to optimize the allocation of lecturer, room, and time resources. The system was developed using the Laravel framework, where the genetic algorithm was used as a solution search engine through truncation selection, one-point crossover, and random resetting mutation mechanisms. Based on a series of parameter tests, the optimal algorithm configuration was obtained at a population size of 150, a maximum generation of 300, a crossover rate of 0.8, and a mutation rate of 0.01. The system implementation proved capable of generating valid schedule solutions without violating constraints (fitness 0) with high computational time efficiency. In a maximum workload scenario of 50 proposals, the system was able to compile a schedule in an average time of 2.717 seconds.**

Keywords*— *Schedule, Scheduling System, Genetics Algorithm, Laravel, Proposal Seminars, Scheduling Optimatization.

I. INTRODUCTION

The final project is a crucial stage in higher education that involves complex coordination between students, supervisors, examiners, and administrators. One of the biggest challenges in its implementation is the process of scheduling proposal seminars and final project hearings. In the D4 Digital Telecommunications Network and D3 Telecommunications Engineering Study Programs at the Malang State Polytechnic, this process involves many variables (lecturers, students, rooms, time) and strict constraints, such as lecturer availability and the suitability of examiners' fields of expertise.

Currently, scheduling is still done manually using spreadsheets (Microsoft Excel). This method has disadvantages, namely that it is time-consuming and has a high risk of human error. A study conducted by Iskandar shows that the duration of manual schedule preparation can take from 1 day for 35 participants to as long as 15 working days when the number of participants reaches hundreds of people [1]. In addition, manual methods often result in schedule conflicts, such as double room usage or conflicting teaching schedules for lecturers, as well as difficulties in accurately matching the expertise of examiners with the topics of student’s theses.

Previous research [2] conducted by Bastian et al. developed a hearing scheduling system at the Faculty of Information Technology, Maranatha Christian University, using the Depth First Search (DFS) method. The system proved to be capable of minimizing schedule conflicts so that hearings ran more smoothly. However, even though DFS is able to provide an exact solution, this algorithm has a fundamental weakness related to computational time efficiency when handling growing datasets (scalability). Referring to the study by Putri, Tulus, and Napitupulu [3], DFS has a time complexity of $\Omega(n!)$, which makes it ineffective for large solution spaces. This limitation is confirmed quantitatively, where there is a drastic increase in execution time with only a small addition of data.

Research conducted by Pangestu et al [4] using Roulette Wheel selection proved effective in resolving schedule conflicts. With a combination of population parameters of 20 and generations of 75, this algorithm was able to produce optimal solutions (fitness 1) on small-scale data in less than 20 seconds, while maintaining time efficiency of under 3 minutes for larger datasets.

Then research by Jeriko et al. [5] developed a web-based scheduling system using genetic algorithms to overcome the inefficiencies of manual methods. Test results proved a significant improvement in performance, with the duration of schedule preparation drastically reduced from an average of 7 days to just 31.26 seconds.

The next study conducted by Wahyuningsih and Helmud [6] applied a genetic algorithm to optimize the allocation of teacher slots and subjects at MTS Negeri 1 Pangkalpinang in order to overcome strict schedule constraints. Their experimental results showed that the algorithm's performance time was greatly influenced by data volume. In tests with a low load (15 subjects), the schedule was successfully generated in 19.56 seconds. However, this duration increased significantly to 471.60 seconds when the data load was increased to 30 subjects.

Aditya and Windha [7] developed an automatic scheduling feature on the GETSMART Learning Management System (LMS) using genetic algorithms. Based on the results of 5 trials,

*Corresponding author

the system successfully generated a class schedule with a very fast average computation time of 7.2 seconds.

Previous research conducted by Fauzi, Akbar, and Kurniawansyah [8] developed a thesis examination scheduling system at Muhammadiyah University Jambi. The research built a system using the PHP programming language and MySQL database to overcome the constraints of limited space and lecturers. The results of the study show that the integration of genetic algorithms into web applications can produce valid and effective schedule recommendations for program chairs, replacing the error-prone manual process.

Guntara et al. [9] also developed a web-based final project hearing scheduling application by applying a genetic algorithm. This research aims to optimize the order of hearings and time allocation by considering various preferences and complex constraints. Performance evaluation results show that the integration of this algorithm into the web platform is capable of providing effective scheduling solutions, with the main advantages being a more even distribution of hearing times and automatic fulfillment of operational constraints.

The next study [10] discusses the development and implementation of a web-based thesis guidance information system for the Informatics Engineering and Computer Engineering Study Programs at Trisakti University. Both systems aim to overcome the inefficiencies of manual thesis management by providing functionalities such as user role management, guidance tracking, and feedback mechanisms. This research uses the waterfall method, utilizing PHP and MySQL for system development, and covers user requirements, system design diagrams, and database structures to improve communication and streamline the thesis process among students, supervisors, and administrative staff.

While various studies have successfully applied Genetic Algorithms for academic scheduling [5][6][7][8][9], a clear research gap remains regarding the optimization of complex constraints beyond basic time and room conflicts. Most previous implementations focus heavily on avoiding hard constraints clashes but often overlook the expertise matching constraint to ensuring the alignment between examiners' expertise and students' research topic which is critical for the quality of the final project examination. Furthermore, distinct performance issues were highlighted in previous studies [6], where the computational efficiency of the algorithm degraded significantly as the dataset size increased.

Unlike previous approaches that rely on standard selection methods like Roulette Wheel [4], this study proposes a differentiated approach by implementing Truncation Selection combined with Elitism Strategy. This specific combination of operators is hypothesised to accelerate the convergence toward an optimal solution (fitness 0) while maintaining computational speed. Consequently, this research focuses on developing a system that is not only valid in avoiding conflicts but also scalable in handling varying data loads and accurate in mapping lecturer expertise.

Based on an analysis of the problems of manual method inefficiency and DFS method scalability limitations, as well as looking at opportunities from previous web-based research, this study proposes the development of a proposal seminar and final project examinations scheduling system using a web-based genetic algorithm using the Laravel framework. This system integrates a genetic algorithm as the main optimization engine, designed not only to handle complex scheduling constraints but also to prioritize the matching of expertise among examiners. Through a modern and structured web technology approach, this research aims to produce an optimal schedule with efficient computation time, as well as provide an interface that is easily accessible to all academic stakeholders.

II. METHOD

A. Genetics Algorithm Design

The genetic algorithm used in this study is designed to solve optimization problems in scheduling Proposal Seminars and Final Project Examinations. In general, the algorithm begins with the initialization of a random schedule population, which is then evaluated based on a fitness function to measure its quality against existing constraints. The best individuals are then selected through a selection process to be developed using genetic operators, namely crossover and mutation, to produce a new, more optimal generation of schedules. The overall system workflow is depicted in Figure 1.

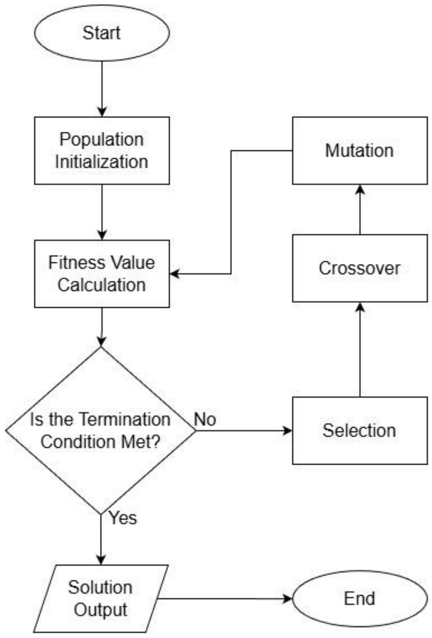


Figure 1. Genetics Algorithm Flowchart

1) Chromosome Representation dan Gene Structure

The chromosome representation scheme is designed using a structured value encoding approach. One individual (chromosome) represents one complete schedule solution that covers the entire list of student proposal seminars for that period. The length of the chromosome is dynamic, depending on the total number of students who register for the hearing or seminar. Each chromosome consists of a series of genes, where each gene represents one seminar schedule slot for a proposal. The position of the gene in the chromosome (locus) functions as an index that refers to the proposal ID, so it is fixed. This aims to ensure the completeness constraint, where each proposal is guaranteed to get one schedule slot. The content of each gene (allele) contains variables that will be optimized by the genetic algorithm. Considering that this system handles two types of academic activities with different resource requirements, the gene structure is divided into two schemes, namely the proposal seminar scheme consisting of 6 alleles:

- Date (D)
- Session (S)
- Room (R)
- Moderator (L_{mod})
- Examiner 1 (L_{ex1})
- Examiner (L_{ex2})

And the final project examination scheme consists of 7 alleles, as follows:

- Date (D)
- Session (S)
- Room (R)
- Supervisor 1 (L_{sup1})
- Supervisor 2 (L_{sup2})
- Examiner 1 (L_{ex1})
- Examiner 2 (L_{ex2})

Mathematically, if C adalah is a chromose and $g_i$is the i-th gene representing the schedule for the i-th proposal, then the chromosome structure can be denoted as:

$$C = \{g_1, g_2, g_3, \dots, g_n\} \tag{1}$$

Where each gene g_i is defined as the following attribute vector for the seminar proposal scheme:

$$g_i = [P_{id}, D_i, S_i, R_i, L_{mod}, L_{ex1}, L_{ex2}] \tag{2}$$

And the following attribute vectors for the seminar proposal scheme:

$$g_i = [P_{id}, D_i, S_i, R_i, L_{sup1}, L_{sup2}, L_{ex1}, L_{ex2}] \tag{3}$$

In the evolutionary process (mutation and crossover), the values of alleles *D, S, R,* and *L* can change stochastically to find the best combination, while Pid remains as a marker of schedule ownership. A visual illustration of the chromosome can be seen in Figure 2. For a visual illustration of the gene structure representation, see Table 1 and Table 2..

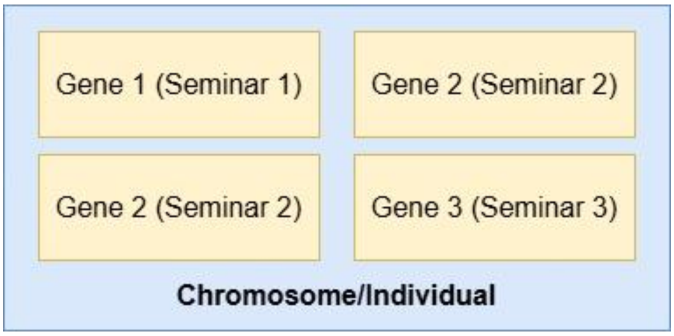


Figure 2. Chromosome Visualization

TABLE I
GENE STRUCTURE VISUALIZATION IN SEMINAR PROPOSAL SCHEDULING

P_{id}	D	S	R	L_{mod}	L_{ex1}	L_{ex2}
1	Jan 1	S1	AH1	Lec A	Lec B	Lec C
2	Jan 2	S2	AH2	Lec M	Lec N	Lec O
3	Jan 3	S3	AH3	Lec X	Lec Y	Lec Z

TABLE II
GENE STRUCTURE VISUALIZATION IN FINAL PROJECT SCHEDULING

P_{id}	D	S	R	L_{sup1}	L_{sup2}	L_{ex1}	L_{ex2}
1	Jan 1	S1	AH1	Lec A	Lec B	Lec C	Lec D
2	Jan 2	S2	AH2	Lec E	Lec F	Lec G	Lec H
3	Jan 3	S3	AH3	Lec I	Lec J	Lec K	Lec L

2) Population Initialization

Population initialization aims to form the first generation containing a set of random schedule solutions. The number of individuals generated depends on the population size parameter input into the system. In this study, population size is one of the test variables (e.g., 50, 100, 150 individuals) to see its impact on computational performance.

The initialization process is carried out using a random assignment mechanism. For each schedule slot of a proposal (locus), the system will randomly select the date, time slot, and room from the input entered by the user and lecturer from the database. This mechanism ensures that each individual in the initial population has a unique and diverse chromosome structure.

For example, if there are 100 proposals and the population size is 50, the system will generate 50 different schedule variations, where each schedule contains a time plot and lecturer for the 100 proposals. The initial population is then evaluated for quality using a fitness function.

3) Fitness Function Evaluation

The fitness function in this study was designed using a penalty minimization approach. Unlike the maximization approach, which seeks the highest value, this system aims to reduce the fitness value to 0. The fitness value represents the total accumulation of penalties (cost) for each constraint violation that occurs in an individual schedule.

The smaller the fitness value (closer to 0), the better the quality of the schedule. A schedule is considered optimal or perfect if it has a fitness value of 0, which means that no constraints are violated. The objective function is formulated as follows:

$$Fitness = \sum_{i=1}^{n} (P_i \times W_i) \tag{4}$$

Description:

- P_i: The number of occurrences of the i-th violation
- W_i: Penalty weight assigned to the i-th type of violation

The system applies a hierarchical penalty weighting mechanism to distinguish between fatal violations and preferences. Details of the penalty weights applied in the algorithm are presented in Table 3.

TABLE III
PENALTY WEIGHT IN FITNESS FUNCTION

Constraint Type	Constraint	Penalty Weight
Hard Constraint	Lecturer Absent / Unavailable	100
Hard Constraint	Lecture Conflict	50
Hard Constraint	Quota Violation	50
Hard Constraint	Room Conflict	50
Hard Constraint	Role Conflict	50
Soft Constraint	Field (Expertise) Mismatch	10

Based on the table above, it can be seen that the system gives the highest priority (100 penalty) to the validity of lecturer availability. This weighting strategy ensures that if the algorithm has to choose between “room schedule conflicts” or “schedules with examiners from the wrong field,” the algorithm will be more tolerant of room conflicts (50 penalty) to be corrected later.

4) *Termination Condition Evaluation*

The termination condition evaluation stage is a control mechanism to determine whether the Genetic Algorithm iteration process should be stopped or continued to the next generation. This process is carried out immediately after the calculation of the fitness value of the entire population is completed in each generation.

In this study, there are two main criteria that trigger the algorithm to stop (stopping criteria):

- Achievement of Optimal Convergence (Fitness = 0): The first and foremost criterion is finding the ideal solution. The system will check the fitness value of the best individual in the current population. If the fitness value of that individual is 0 (zero), the algorithm will immediately stop. This value of 0 indicates that the system has found a valid schedule arrangement without any violations of hard constraints (no schedule, room, or lecturer conflicts). This solution is then set as the final scheduling result.
- Reaching the Maximum Generation Limit: If a solution with a fitness of 0 has not been found, the algorithm will check the number of generations that have been passed. If the current iteration has reached the Maximum Generation limit that was determined at the beginning of the parameters, then the process will be forcibly stopped. Individuals with the best fitness values in the last generation will be taken as the best approximate solution, even though they may still contain soft constraint violations.

5) *Selection Method*

The selection stage aims to select individuals from the current population to be used as parents to produce the next generation. In this study, the method applied is Truncation Selection combined with the Elitism strategy.

Truncation Selection is a selection method used to select parents from a large set of chromosomes by taking a portion of individuals from the population to be used as parents[11]. Elitism is the process of making copies of a set of individuals with the best fitness values from one generation to the next. Elitism aims to preserve individuals with the best fitness values so that they do not become extinct during the evolutionary process [12].

The Truncation Selection method works by sorting all individuals in the population based on their fitness values in ascending order, from the lowest to the highest. Once the population is sorted, the system sets a selection threshold by taking the top third (1/3) of the total population. The individuals in this top third group are considered to have chromosome quality closest to the optimal solution and are eligible to enter the mating pool.

These selected individuals will then act as parents to undergo the crossover and mutation process to produce new offspring. Meanwhile, the remaining 2/3 of individuals with lower fitness values will be eliminated and not involved in the regeneration process. This approach is chosen to provide strong selection pressure, so that the algorithm can converge more quickly towards a valid schedule solution and minimize constraint violations.

Along with this process, the Elitism strategy is applied to ensure convergence stability. Individuals with the highest absolute fitness values in the current generation will be copied directly to the next generation without undergoing genetic modification. This strategy serves as a safeguard to ensure that the best schedule solutions that have been found will not be lost or damaged due to crossover or mutation operations that are stochastic in nature.

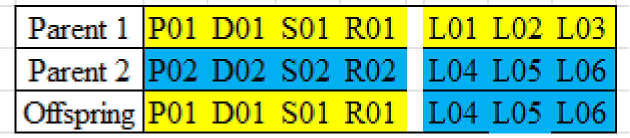
6) *Crossover*

Crossover is a process in which two chromosomes are randomly selected based on a specific probability to exchange gene fragments at the same position [13]. Crossover aims to explore the search space for solutions by creating new genetic variations. This process involves the exchange of genetic material (genes) between two parent individuals that have been selected in the previous selection stage, in order to produce offspring that inherit characteristics from both parents.

In this study, the crossover process is not performed on all parent pairs absolutely, but is controlled by the crossover probability parameter (P_c). For each parent pair, the system will generate a random number r between 0 and 1. If the value of r is less then P_c ($r < P_c$), then the crossover process will be executed. Conversely, if $r \geq P_c$, then both parents will be copied directly into offspring without genetic changes.

The method applied is One-Point Crossover. The mechanism works by randomly determining a cut-point on the chromosome length [14]. Genes located to the right of the cut-point in the first parent will be exchanged with genes in the same position in the second parent, and vice versa.

In the context of this scheduling system, the gene exchange represents the combination of time, space, or lecturer arrangements from two different schedules. This is expected to combine the optimal parts of one parent's schedule with the optimal parts of its partner's schedule, resulting in a new schedule with better fitness quality.



Parent 1	P01	D01	S01	R01	L01	L02	L03
Parent 2	P02	D02	S02	R02	L04	L05	L06
Offspring	P01	D01	S01	R01	L04	L05	L06

Figure 3. Ilustraion of One-Point Crossover

7) *Mutation*

After the crossover process is complete, the next stage is mutation. Mutation is a genetic operator that works by randomly changing one or more gene values in a chromosome [15]. The main purpose of mutation is to maintain genetic diversity in the population and prevent the algorithm from premature convergence on local solutions. In the context of scheduling, mutation plays an important role in breaking scheduling configuration deadlocks that may not be resolved through the crossover process alone.

In this study, the mutation method applied is Attribute-Level Random Resetting. Unlike conventional mutations that replace one chromosome block (one complete schedule) at a time, this system checks the mutation probability on each attribute that makes up the schedule independently. The attributes that are the targets of mutation include: Room, Date, Session, Examiner 1, and Examiner 2.

The mutation mechanism is implemented by generating a random number r with a range of 0-1 for each attribute in a gene. If the value of r is smaller than the specified Mutation Rate (Pm), then the value of that attribute will be replaced. This approach allows for partial improvements to the schedule without having to overhaul other valid schedules within a gene.

B. *System Design*

The system design phase aims to provide a clear specification of the structure and behavior of the software to be built. The discussion in this section is divided into two main components. First, database design, which defines the table structure and data relationships to support the needs of Genetic Algorithms. Second, the system workflow, which describes the user interaction mechanism and data processing logic from start to finish to produce a valid schedule. The following is a description of each of these designs.

1) *Database Design*

The database design in this study is focused on supporting the computational needs of Genetic Algorithms, particularly in representing chromosome and gene structures. The database not only functions as a medium for storing administrative data, but also as a provider of input parameters (such as space and time) and constraints that must be obeyed by the scheduling system. The table structure is designed so that the system can retrieve data efficiently during the population initialization and fitness evaluation processes. The design of the relationship between entities used can be seen in Figure 4.

Figure 4 illustrates the interconnection between data that supports the scheduling mechanism. The structure centers on the proposal table as the entity to be scheduled. The lecturer, room, and time attribute tables function as a set of resources to be combined by the genetic algorithm.

The relationships formed, such as the relationship between dosen (lecturers) table and bidang_minat (fields of interest) as

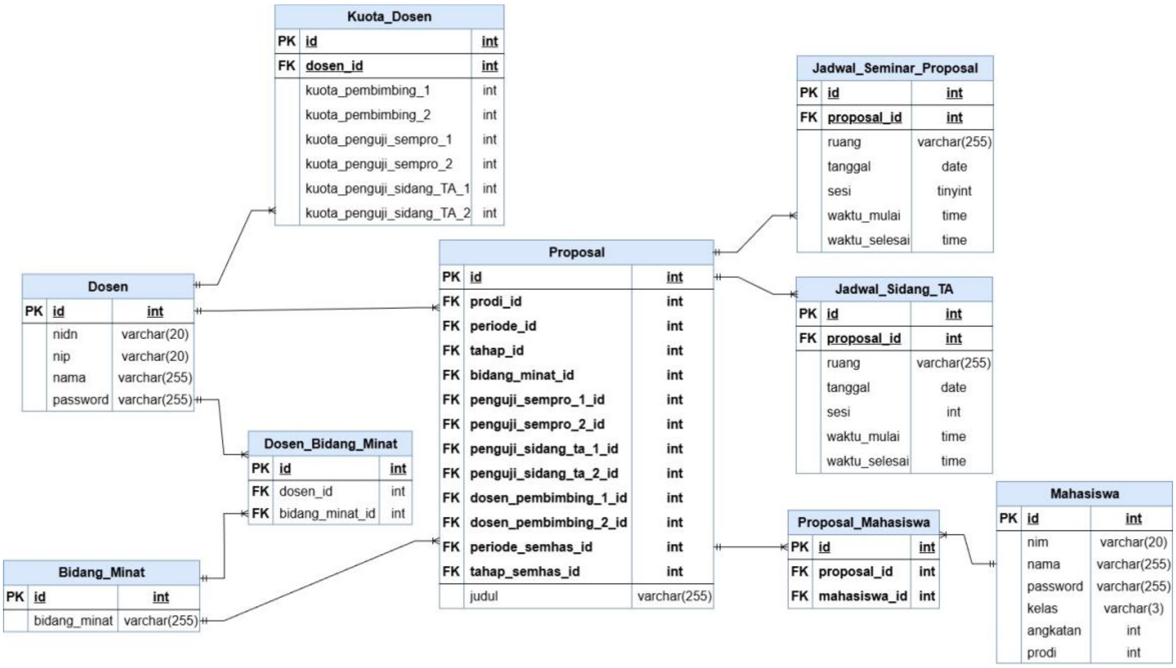


Figure 4. Entity Relationship Diagram

well as kuota_dosen (lecturer quotas), are designed to facilitate constraint checking when the fitness function is working. Meanwhile, the jadwal_seminar_proposal (proposal seminar schedule) and jadwal_sidang_ta (final project examination schedlue) table serves to store the best schedule solutions (best individual) that have been generated, including information on time, location, and validated examination teams.

2) Scheduling System Workflow

The main business process of the system begins with the initialization of scheduling parameters by the user. At this stage, the user enters constraint variables, which include room availability, the range of implementation dates, time slot distribution, and a list of lecturer unavailability times. In addition, the user also determines the academic period and stage as data filter keys.

After the input parameters are validated, the system automatically retrieves data from the database. The system downloads student proposal data that matches the selected period, as well as complete lecturer data with guidance/test quota attributes and areas of interest. This data set is then converted into an initial population to be processed by the Genetic Algorithm engine.

The results of the algorithm execution in the form of the best schedule (best fitness) are then stored back into the database (store). Finally, the system presents a visualization of the trial schedule to the user interface in a structured table format, making it easier for users to perform final verification.

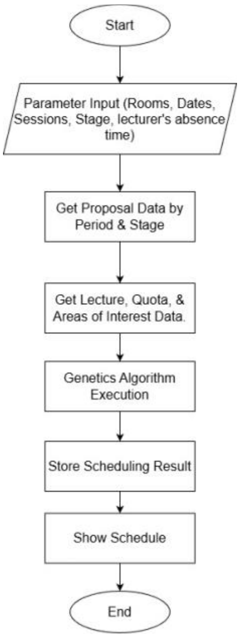


Figure 5. Scheduling System Flowchart

C. Testing Scenario

1) Testing Environment

The testing environment was established to evaluate the computational performance of the Genetic Algorithm under controlled conditions. To ensure the reliability and consistency of the experimental data, all test scenarios were executed on a single, dedicated computing device. This approach was adopted to eliminate potential discrepancies caused by hardware variations, particularly in measuring critical metrics such as execution time and memory consumption. The specific hardware and software configurations used throughout the testing process are comprehensively detailed in Table IV.

TABLE IV
TEST ENVIRONMENT SPECIFICATIONS

Component	**Specifications**
Processor	Intel Core i3-6006U @ 2.00GHz
Memory (RAM)	12 GB
Operating System	Windows 10
Programming Language	PHP 8.2.12
Database	MySQL 8.4

2) Testing Dataset

The test dataset was constructed based on historical data from the D3 Telecommunications Engineering Study Program and the D4 Digital Telecommunications Network Study Program at Polinema for the 2023/2024 academic year. The test data volume was set at 50 proposals. This number was chosen based on observations of the average number of seminar and hearing participants per stage in both study programs, which generally ranged from 20 to 30 to a maximum of 50 proposals, so this number was considered representative for testing the system's workload.

The availability of resources, including the number of rooms, days, and sessions, was determined based on the average real facilities allocated by the study program for each seminar..

TABLE V
TESTING DATASET DETAILS

Data Parameter	**Amount of Data**
Number of Lectures	34
Number of Rooms	5
Number of Sessions per Day	4
Number of Days Available	3
Number of Proposals	50

3) Parameter Testing Scenario

Parameter testing was conducted to find the combination of Genetic Algorithm parameter values that produced the most optimal schedule solution with efficient execution time. The parameters tested include population size, generation limit, crossover rate, and mutation rate. The testing method was carried out in stages by varying the value of one test parameter and setting a constant value for the other parameters. Details of the parameter value scenarios tested can be seen in Table VI. Each parameter value was tested 10 times to obtain a valid average value.

TABLE VI
TEST ENVIRONMENT SPECIFICATIONS

No	Tested Parameter	Variable Value	Constant Parameter
1	Population Size	50, 100, 150, 200, 250, 300	Max Gen: 100, P_m: 0.05, P_c: 0,8
2	Maximum Generation	100, 200, 300, 400, 500	Pop Size: 100, P_m: 0.05, P_c: 0,8
3	Mutation Rate (P_m)	0.01, 0.05, 0.1, 0.2	Pop Size: 100, Max Gen: 100, P_c: 0,8
4	Crossover Rate (P_c)	0.7, 0.8, 0.9, 0.95	Pop Size: 100, Max Gen: 100, P_m: 0.05

4) Computation Time Testing Scenario

Computational time testing was conducted to measure algorithm efficiency and system scalability when handling varying data volumes. Execution time (running time) was calculated from the population initialization stage until the system successfully found a scheduling solution that met the termination conditions.

In this scenario, the genetic algorithm parameter configuration (population size, number of generations, mutation rate, and crossover rate) was set using a combination of optimal values obtained from the previous parameter testing stage. The independent variable manipulated was the input data load, namely the number of final project proposals that had to be scheduled. The testing was conducted using three variations of data: 20, 30, and 50 proposals. The selection of these variations was based on observations of the average number of seminar participants and final project hearings per stage in the D3 Telecommunications Engineering and D4 Digital Telecommunications Network Study Programs at Polinema, so that the test results could represent the system's performance under actual workload conditions.

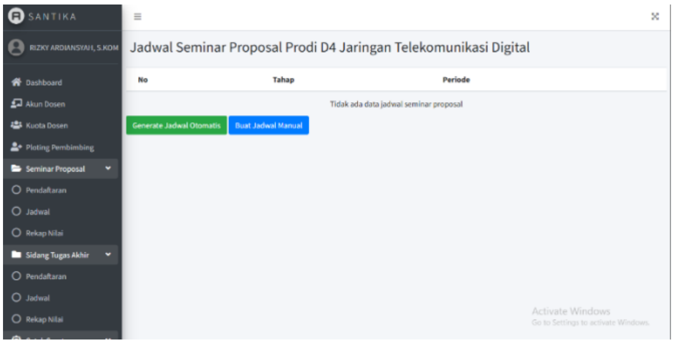
The purpose of this test is to analyze the characteristics of processing time growth with increasing data load. Given the stochastic nature of Genetic Algorithms, each test scenario will be run 10 times to obtain a valid average execution time and minimize bias due to processor performance fluctuations.

III. RESULTS AND DISCUSSION

This chapter details the system implementation and testing results, covering the user interface, Genetic Algorithm parameter optimization, and computational time efficiency analysis.

A. System Implementation Results

The results of system implementation are the technical realization of the database design and system workflow that were defined in the previous stage. The scheduling system is built on a web-based platform to provide an interactive interface for users to manage the scheduling process. The following is a visualization of the main interface of the system, which represents the operational flow of the application, from data input to the presentation of schedule results that have been validated by the algorithm.

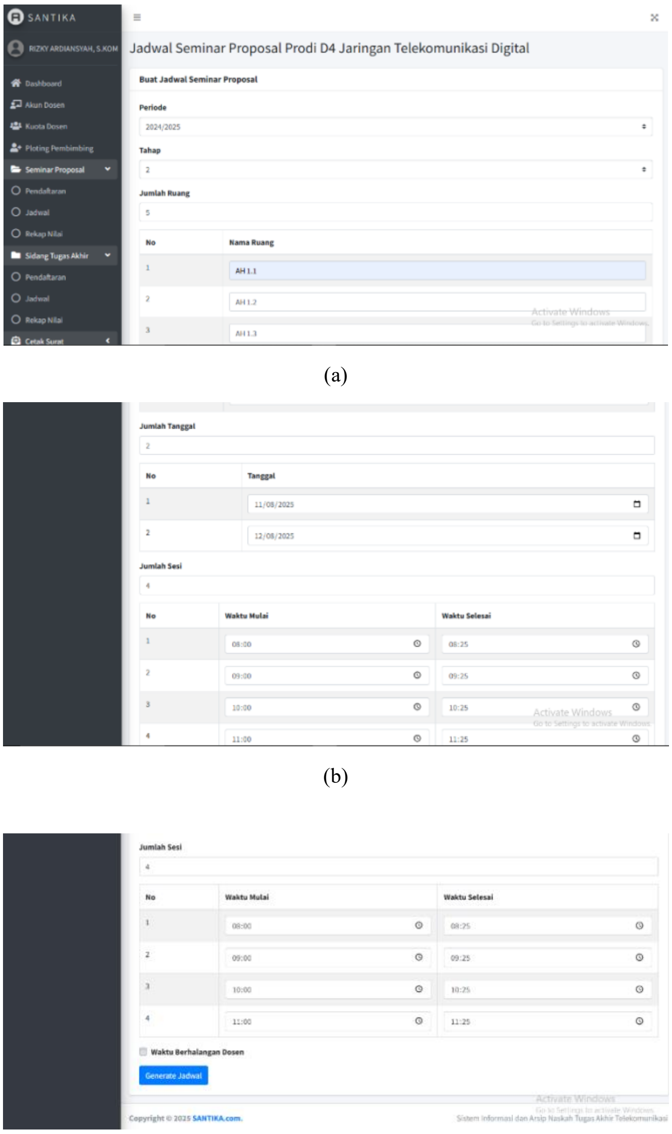


Jadwal Seminar Proposal Prodi D4 Jaringan Telekomunikasi Digital

No	Tahap	Periode

Tidak ada data jadwal seminar proposal

Figure 6. Schedulling Homepage



Jadwal Seminar Proposal Prodi D4 Jaringan Telekomunikasi Digital

Buat Jadwal Seminar Proposal

Periode: 2024/2025

Tahap: 2

Jumlah Ruang: 5

No	Nama Ruang
1	AH 1.1
2	AH 1.2
3	AH 1.3

(a)

Jumlah Tanggal: 2

No	Tanggal
1	11/08/2025
2	12/08/2025

Jumlah Sesi: 4

No	Waktu Mulai	Waktu Selesai
1	08:00	08:25
2	09:00	09:25
3	10:00	10:25
4	11:00	11:25

(b)

Jumlah Sesi: 4

No	Waktu Mulai	Waktu Selesai
1	08:00	08:25
2	09:00	09:25
3	10:00	10:25
4	11:00	11:25

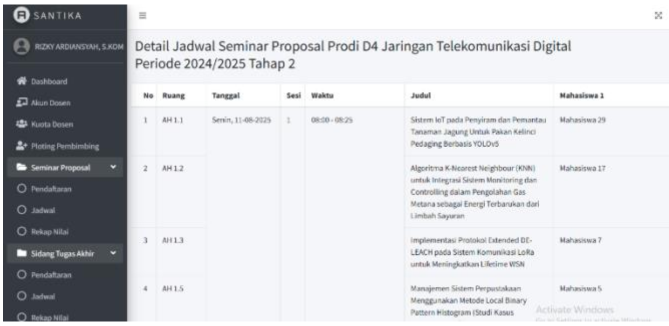
☐ Waktu Berhalangan Dosen

Copyright © 2025 SANTIKA.com.

Sistem Informasi dan Arsip Naskah Tugas Akhir Telekomunikasi

(c)

Figure 7. Initialization Page Scheduling Parameters: (a) Period and Room Settings; (b) Date and Session Settings; (c) Schedule Generation Execution



Detail Jadwal Seminar Proposal Prodi D4 Jaringan Telekomunikasi Digital Periode 2024/2025 Tahap 2

No	Ruang	Tanggal	Sesi	Waktu	Judul	Mahasiswa 1
1	AH 1.1	Senin, 11-08-2025	1	08:00 - 08:25	Sistem IoT pada Penyiram dan Pemantau Tanaman Jagung Untuk Pakan Kelinci Pedaging Berbasis YOLOv5	Mahasiswa 29
2	AH 1.2				Algoritma K-Nearest Neighbour (KNN) untuk Integrasi Sistem Monitoring dan Controlling dalam Pengolahan Gas Metana sebagai Energi Terbarukan dari Limbah Sayuran	Mahasiswa 17
3	AH 1.3				Implementasi Protokol Extended DE-LEACH pada Sistem Komunikasi LoRa untuk Meningkatkan Lifetime WSN	Mahasiswa 7
4	AH 1.5				Manajemen Sistem Perpustakaan Menggunakan Metode Local Binary Pattern Histogram (Studi Kasus	Mahasiswa 5

Figure 8. Schedule Generation Results

The final results of the Genetic Algorithm execution are presented in Figure 8, which displays a list of automatically generated seminar proposal schedules. In this interface, the system presents the best schedule solution (best individual) in a table format that maps each student proposal title to a specific time slot (date and session) and room. As seen in the "Room" and "Time" columns, the system successfully allocated several proposals to different rooms at the same time (parallel) without any conflicts, indicating that the algorithm has successfully met the main constraints (hard constraints) related to resource and time usage.

B. Parameter Testing Results

The genetic algorithm parameters were evaluated to identify the configuration that yields the optimal fitness value (i.e., a penalty value of 0) with efficient computational time.

TABLE VII
POPULATION SIZE PARAMETER TEST RESULTS

No	Population Size	Fitness Value	Computation Time (s)
1	50	2	2.893
2	100	5	4.832
3	150	1	4.929
4	200	3	7.377
5	250	5	10.81
6	300	6	12.576

TABLE VIII
MAXIMUM GENERATION PARAMETER TEST RESULTS

No	Maximum Generation	Fitness Value	Computation Time (s)
1	100	4	3.439
2	200	0	3.264
3	300	0	3.243
4	400	0	3.602
5	500	0	3.705

TABLE IX
MUTATION RATE PARAMETER TEST RESULTS

No	Mutation Rate	Fitness Value	Computation Time (s)
1	0.01	0	0.957
2	0.05	3	2.986
3	0.1	23	8.049
4	0.2	36	12.786

TABLE X
CROSSOVER RATE PARAMETER TEST RESULTS

No	Crossover Rate	Fitness Value	Computation Time (s)
1	0.7	6	5.189
2	0.8	4	4.177
3	0.9	4	4.311
4	0.95	5	4.131

Based on the experimental data presented in Table 7, testing population variation shows a trade-off between solution quality and time. A population of 150 produces the lowest average penalty, but requires almost twice the execution time compared to a population of 50. Therefore, by prioritizing schedule validity (smallest fitness value) as the main criterion for system performance, a population size of 150 is set as the optimal parameter.

Table 8 shows that the algorithm began to reach optimal convergence (fitness value of 0 or no violations) in the 200th generation. Adding generations above 300 (up to 500) proved ineffective because the fitness value remained stagnant at 0, while the computation time continued to increase. Based on this data, a maximum number of 300 generations is the most optimal parameter value in this case study.

Testing shows that the system is highly sensitive to the mutation rate. A small mutation rate (0.01) actually produces the most perfect solution (fitness value of 0) with the fastest time (0.9 seconds). Conversely, a high mutation rate (0.2) caused the penalty value to spike to 36 and the computation time to slow down drastically to 12.7 seconds. This indicates that too high a mutation rate is destructive and damages the structure of the schedule chromosome that has been formed.

Based on the tested value variations, a crossover rate of 0.8 provided the most stable results with an average penalty value of 4. Although it did not reach a perfect value of 0 in this partial test, the value of 0.8 showed better performance than the extreme value of 0.95, which actually increased the average penalty.

Based on this analysis, the optimal parameter configuration set for this system is a population size of 150, a maximum generation of 300, a mutation rate of 0.01, and a crossover rate of 0.8**.**

C. Computation Time Testing Results

The final stage of the testing series is the analysis of computing time performance. This test uses control parameters that have been tested to produce the best scheduling solution. Table 5 shows the average duration of time required for the system to complete scheduling at various data load levels.

TABLE XI
TEST ENVIRONMENT SPECIFICATIONS

No	Number of Proposals	Computation Time (s)	Fitness Value
1	20	0.283	0
2	30	0.583	0
3	50	2.717	0

The experimental results presented in Table 11 provide critical insights into two key performance metrics: computational efficiency and solution quality. Regarding efficiency, the system demonstrates high responsiveness, particularly in lower data ranges. For datasets containing 20 and 30 proposals, the algorithm achieved convergence in sub-second durations of 0.283 seconds and 0.583 seconds, respectively. A noticeable shift occurs at the maximum load of 50 proposals, where the execution time increases to 2.717 seconds due to the non-linear complexity of the scheduling problem.

Beyond speed, the most significant finding in Table 11 is the consistency of the Fitness Value, which remains at 0 across all workload variations. A fitness value of 0 signifies that the resulting schedules are "perfect" in the context of the defined constraints; meaning every proposal is assigned a valid slot, no lecturer is double-booked, and room capacities are respected. This confirms that the increase in processing time (up to 2.7 seconds) is effectively utilized by the algorithm to resolve conflicts and eliminate penalties, rather than returning a premature, invalid solution.

From a practical perspective, this performance is highly superior to the manual scheduling method. A processing time of under 3 seconds to generate 50 conflict-free schedules implies that the system operates in near real-time. Consequently, the system is validated as a robust solution for routine academic operations, offering a balance of high-speed processing and zero-error accuracy.

IV. CONCLUSION

This research has successfully developed a web-based seminar proposal and final project examination scheduling system using the Laravel framework integrated with Genetic Algorithms as an optimization engine. The application of this algorithm has proven effective in solving complex academic resource allocation problems, where the optimal parameter configuration in this case study was obtained at a population size of 150, a maximum generation of 300, a mutation rate of 0.01, and a crossover rate of 0.8. Based on scalability testing results with a maximum workload of 50 proposals, the system showed highly efficient computational performance with the ability to generate conflict-free schedules (fitness 0) in an average time of 2.717 seconds. With all main constraints (hard constraints) met and high time efficiency, this system is declared valid and feasible for implementation in the D3 Telecommunications Engineering and D4 Digital Telecommunications Network Study Programs at the Malang State Polytechnic.

REFERENCES

[1] A.P.S. Iskandar, "Optimasi Penjadwalan Ujian Tugas Akhir Dengan Menggunakan Algoritma Genetika (Final Project Scheduling Optimization Using Genetic Algorithm)," *J-COSINE*, vol. 5, no. 1, pp. 40–48, Jun. 2021, [Online]. Available: http://jcosine.if.unram.ac.id/

[2] Y. Bastian, D. J. Surjawan, and A. Adelia, "Aplikasi Penjadwalan Sidang di Fakultas Teknologi Informasi UK Maranatha," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 1, Apr. 2018, doi: 10.28932/jutisi.v4i1.760.

[3] S. Eka Putri and N. Napitupulu, "Implementation and Analysis of Depth-First Search (DFS) Algorithm for Finding The Longest Path," the International Seminar on Operational Research (InteriOR), Medan, Indonesia, Aug. 2011, doi: 10.13140/2.1.2878.2721

[4] L. A. Pangestu, S. H. Suryawan, and A. J. Latipah, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran," *Jurnal Informatika*, vol. 10, no. 2, pp. 194–205, Oct. 2023, doi: 10.31294/inf.v10i2.16701.

[5] T. Kristian Jeriko, D. Faizal Racma, C. Ety Widjayanti, and A. Ary Setyawan, "Penerapan Algoritma Genetika Dalam Sistem Informasi Penjadwalan Mata Kuliah Berbasis Website Pada STIKOM Yos Sudarso Purwokerto," vol. 6, no. 1, pp. 101–118, 2022, doi: 10.24912/jmstkik.v6i1.17262.

[6] D. Wahyuningsih and E. Helmud, "Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan pada MTS Negeri 1 Pangkalpinang," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 3, pp. 435–441, Dec. 2020, doi: 10.32736/sisfokom.v9i3.994.

[7] L. Aga Aditya and W. P. Mega, "Implementasi Algoritma Genetika Untuk Penjadwalan Mata Pelajaran Pada LMS Getsmart," 2017. [Online]. Available: http://putranugraha.id.

[8] M. Fauzi, Z. Akbar, and K. Kurniawansyah, "Sistem Penjadwalan Sidang Skripsi Dengan Algoritma Genetika Pada Program Studi Informatika Universitas Muhammadiyah Jambi," *Jurnal Informatika, Sistem Informasi dan Kehutanan (FORSINTA)*, vol. 3, no. 2, 2024.

[9] Rangga Gelar Guntara, M. Rizki Nugraha, Y. Prasetyo, and R. Aprilia, "Implementasi Algoritma Genetika Untuk Aplikasi Penjadwalan Sidang Tugas Akhir Berbasis Web," *Jurnal Minfo Polgan*, vol. 12, no. 2, pp. 2224–2232, Nov. 2023, doi: 10.33395/jmp.v12i2.13206.

[10] D. Yunda and S. Sari, "Sistem Informasi Tugas Akhir Berbasis Web di Jurusan Teknik Informatika Universitas Trisakti," vol. 2, pp. 73–86.

[11] S. D. Immanuel and U. Kr. Chakraborty, "Genetic Algorithm: An Approach on Optimization," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, 2019, pp. 701–708. doi: 10.1109/ICCES45898.2019.9002372.

[12] A. Mauko, A. Fanggidae, and Y. Polly, "Analysis Of Elitism In Genetic Algorithm Using Ordinal Representation Coding On Traveling Salesman Problems," *J-Icon : Jurnal Komputer dan Informatika*, vol. 10, no. 2, Dec. 2022, doi: 10.35508/jicon.v10i2.8473.

[13] Q. Zhang, "An optimized solution to the course scheduling problem in universities under an improved

genetic algorithm," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 1065–1073, 2022, doi: doi:10.1515/jisys-2022-0114.

[14] A. M. Fajrin, "Analisis Performa Dari One-Point, Multi-Point Dan Order Crossover Di Algoritma Genetika," *semanTIK*, vol. 7, no. 2, pp. 1–5, 2021, doi: 10.5281/zenodo.5793087.

[15] R. Kumar, M. Memoria, A. Gupta, and M. Awasthi, "Critical Analysis of Genetic Algorithm under Crossover and Mutation Rate," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 976–980. doi: 10.1109/ICAC3N53548.2021.9725640.