

SISTEM PENJADWALAN SIDANG TUGAS AKHIR MENGGUNAKAN ALGORITMA GENETIKA

Berlian Bayu Aji¹, Susetyo Bagas Bhaskoro², Siti Aminah³

^{1,2,3} Teknologi Rekayasa Otomasi, Politeknik Manufaktur Bandung

Email: ¹berli.bayu2@live.com, ²bagas@ae.polman-bandung.ac.id, ³siti@ae.polman-bandung.ac.id

Abstrak

Objektif dari penelitian ini adalah membuat penjadwalan sidang tugas akhir mahasiswa secara otomatis menggunakan algoritma genetika pada Program Studi Teknologi Rekayasa Otomasi Politeknik Manufaktur Bandung. Algoritma genetika (GA) adalah sebuah metode heuristik atau metode pencarian nilai optimal dengan menggunakan prinsip-prinsip evolusi. GA ini digunakan sebagai algoritma pencarian solusi pada penelitian ini. Solusi tersebut diimplementasikan menjadi bentuk program yang dapat membuat jadwal secara otomatis berdasarkan batasan-batasan yang ditentukan oleh pengguna program. Parameter penjadwalan yang difokuskan pada penelitian ini adalah masalah duplikasi atau bentrok dan kesesuaian kategori antara tugas akhir dengan penguji. Berdasarkan parameter tersebut program dapat menghasilkan jadwal yang optimal dengan tingkat keberhasilan yang bervariasi bergantung pada rasio antara jumlah penguji sidang dengan jumlah kebutuhan penguji pada satu waktu, dimana rasio di atas 2:1 menghasilkan keberhasilan di atas 90% untuk kasus duplikasi, dan rasio di atas 1,5:1 untuk kasus kesesuaian kategori.

Kata Kunci: Algoritma Genetika, Penjadwalan, Gen, Kromosom, Fitness

Abstract

Objective this research is to automatically schedule student final assignments using genetic algorithms in the Automation Engineering Technology Study Program of the Manufacturing Polytechnic in Bandung. Genetic Algorithm (GA) is a heuristic method or a method of finding optimal values using the principles of evolution. GA is used as a solution search algorithm in this research. The solution was implemented into a program that can create a schedule automatically based on the limitations set by program users. Scheduling parameters that are focused on in this study are duplication or clash issues and category suitability between the final project and examiners. Based on these parameters the program can produce the optimal schedule with varying success rates depending on the ratio between the number of trial examiners and the number of examiners needed at one time, where a ratio above 2: 1 result in success above 90% for duplication cases, and the ratio is above 1.5: 1 for category conformance cases.

Keywords: Genetic Algorithms, Scheduling, Gene, Chromosome, Fitness

I. PENDAHULUAN

Penjadwalan adalah suatu proses pengalokasian sebuah kegiatan dalam jangka waktu tertentu dan kapan sesuatu tersebut seharusnya terjadi (Pinedo, 2016). Pada penelitian *Generating university course timetable using Genetic Algorithms and local search*

(Abdullah dan Turabieh, 2008) menyebutkan bahwa masalah penjadwalan perkuliahan di universitas adalah masalah yang sangat sulit diselesaikan dengan menggunakan metode tradisional, karena jumlah perhitungan yang diperlukan untuk menemukan solusi optimal selalu meningkat secara

eksponensial disesuaikan dengan ukuran masalah yang juga selalu meningkat.

Penelitian ini mulai dilakukan pada tahun ajaran 2019/2020 di Program Studi Teknologi Rekayasa Otomasi Politeknik Manufaktur Bandung, dimana pada saat itu Program Studi memiliki jumlah mahasiswa tingkat akhir sebanyak 55 mahasiswa. Aturan pelaksanaan ujian di Program Studi dilakukan pada periode waktu tertentu yang telah dijadwalkan, oleh karena itu waktu pelaksanaan ujian, ketersediaan pengujian dan ketersediaan ruangan ujian juga menjadi perhatian sehingga tidak terjadi duplikasi waktu, pengujian dan ruangan di waktu yang bersamaan.

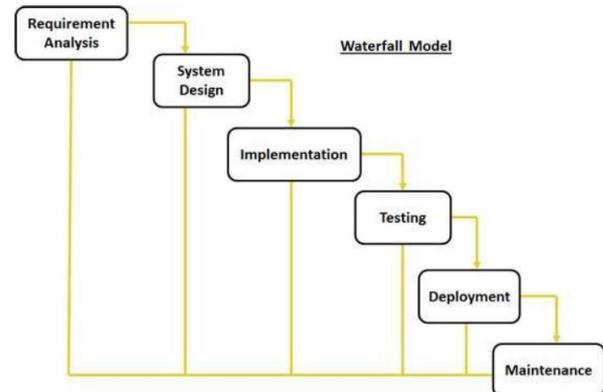
Sebagai upaya menyelesaikan kendala yang terjadi, pada penelitian ini mencoba melakukan rujukan penelitian terkait dengan proses penjadwalan yang dilakukan secara otomatis (Mawaddah dan Mahmudy, 2006) dan (Abdullah dan Turabieh, 2008). Berdasarkan dari referensi yang ada, algoritma genetika (GA) menjadi metode yang dipilih untuk menjawab permasalahan yang terjadi pada penelitian yang sedang dilakukan.

Beberapa penelitian juga menggunakan GA untuk membuat proses penjadwalan secara otomatis, antara lain Sari (Sari dkk., 2019) yang menggunakan parameter waktu untuk menggambarkan GA mampu melakukan otomatisasi penjadwalan. Begitu juga penelitian yang dilakukan oleh Permadi (Permadi, 2010), melakukan perbandingan parameter crossover pada GA.

Penelitian Fiarni (Fiarni dkk., 2015) menggunakan *forward chaining method* pada penjadwalan sidang yaitu algoritma pencarian solusi secara berantai. Penelitian ini menghasilkan alokasi waktu yang cocok (tidak disertai dengan ruangan) dari sebuah data mahasiswa dengan kelompok dosen yang menjadi pengujian mahasiswa tersebut. Perbedaan antara *Forward Chaining Method* dengan Algoritma Genetika (GA) sebagai solusi pada masing-masing penelitian di atas adalah, ketika *Forward Chaining Method* memerlukan data mahasiswa dengan pengujinya, sehingga untuk 55 orang mahasiswa digenerasi sebanyak 55 jadwal dengan ketentuan masing-masing mahasiswa sudah memiliki data pengujian sidang. GA dapat menghasilkan satu jadwal yang dapat mengakomodir keseluruhan komponen seperti mahasiswa, pengujian dan ruangan, dengan hanya menentukan batasan-batasan penjadwalan tersebut, sehingga GA dipilih sebagai solusi dalam penelitian yang sedang dilakukan.

II. METODOLOGI PENELITIAN

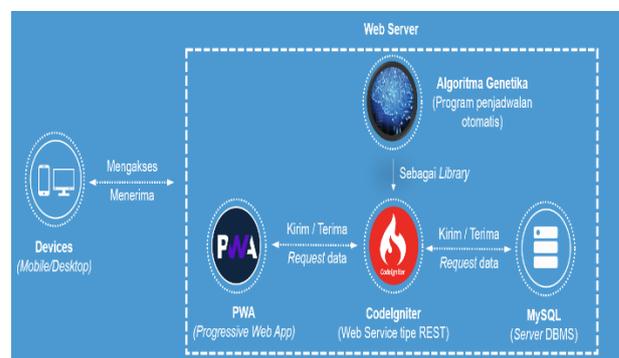
Perancangan pada penelitian ini menggunakan metode *waterfall* (Bulman, 2017). Gambar 1 adalah metodologi yang digunakan.



Gambar 1. Tahapan Metodologi *Waterfall*

Arsitektur Sistem

Sistem yang dikembangkan berbasis web dengan mengimplementasikan konsep programming *fullstack*. Dalam pengembangannya aplikasi ini dibagi ke dalam dua bagian yaitu *back-end*, dan *front-end*. Pada penelitian ini bagian *back-end* perangkat lunak adalah web server yang dibuat dalam bentuk REST API (*Representational State Transfer Application Programming Interface*) dan bagian *front-end*. Pembuatan perangkat lunak tersebut mengikuti arsitektur yang ditunjukkan pada Gambar 2 di bawah ini.



Gambar 2. Arsitektur Sistem yang Dibuat

Berdasarkan Gambar 2 di atas, bagian yang menjadi fokus pembahasan utama dalam penelitian ini adalah pembuatan program GA yang berfungsi sebagai program penjadwalan otomatis.

Model Penjadwalan

Ketika melakukan analisis kebutuhan, penelitian ini mengasumsikan kasus penjadwalan menjadi poin-poin berikut:

1. Kelompok mahasiswa terbagi atas mahasiswa kelas 4 AED dengan jumlah 23 orang dan mahasiswa kelas 4 AEE dengan jumlah 32 orang.
2. Sidang tugas akhir hanya dilakukan pada ruangan dan slot waktu tertentu yang sudah disediakan oleh panitia tugas akhir.
3. Pelaksanaan sidang tugas akhir melibatkan satu mahasiswa, dua pembimbing, dan empat orang penguji dalam satu slot ruangan dan waktu.

Berdasarkan model penjadwalan yang diasumsikan pada tiga poin di atas, terdapat komponen-komponen yang ada pada kasus penjadwalan tersebut. Komponen dibagi ke dalam tiga kategori yaitu konstanta, himpunan, dan batasan-batasan penjadwalan.

Berikut adalah komponen-komponen pada model kasus yang digunakan dalam penelitian ini dan berupa konstanta:

1. R = Jumlah ruangan
2. T = Jumlah slot waktu perhari
3. D = Jumlah hari sidang tugas akhir
4. P = Jumlah penguji per ruangan

Berikut adalah komponen-komponen pada model kasus yang digunakan dalam penelitian ini dan berupa himpunan:

1. TA = Himpunan judul tugas akhir mahasiswa yang dinyatakan sebagai berikut, $TA = \{ta_1, ta_2, \dots, ta_n\}$, masing-masing dari anggota himpunan tersebut mewakili satu judul tugas akhir mahasiswa.
2. D = Himpunan dosen yang dinyatakan sebagai berikut, $D = \{dsn_1, dsn_2, \dots, dsn_n\}$, masing-masing dari anggota himpunan tersebut mewakili satu nama dosen.
3. Kt = Himpunan kategori tugas akhir yang dinyatakan sebagai berikut, $Kt = \{sw, hw\}$, dimana sw mewakili *software* dan hw mewakili *hardware*.

Masing-masing dari himpunan tersebut memiliki relasi untuk menyatakan hubungan antara keduanya, relasi mengakibatkan pada pembentukan himpunan baru.

1. MD adalah relasi antara judul tugas akhir mahasiswa dengan dosen, yang merupakan dosen pembimbing dari tugas akhir mahasiswa tersebut, dinyatakan sebagai berikut:

$$MD = \{[ta_1, (ds_1, ds_2)], [ta_2, (ds_3, ds_4)], \dots, [ta_n, (ds_1, ds_3)]\}$$

2. MP adalah relasi antara judul tugas akhir mahasiswa dengan dosen, yang merupakan dosen penguji sidang dari tugas akhir mahasiswa tersebut, dinyatakan sebagai berikut:

$$MP = \{[ta_1, (ds_1, ds_2, ds_3)], [ta_2, (ds_3, ds_4, ds_5)], [ta_n, (ds_1, ds_3, ds_4)]\}$$

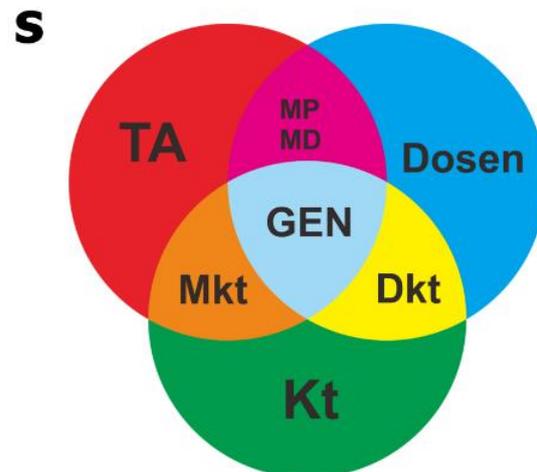
3. MKt adalah relasi antara judul tugas akhir mahasiswa dengan kategori tugas akhir, dinyatakan sebagai berikut:

$$MKt = \{[ta_1, sw], [ta_2, sw], \dots, [ta_n, hw]\}$$

4. Dkt adalah relasi antara penguji dengan kategori yang mengindikasikan spesialisasi, dinyatakan sebagai berikut:

$$DKt = \{[ds_1, sw], [ds_2, sw], \dots, [ds_n, hw]\}$$

Relasi digambarkan seperti pada Gambar 3 di bawah ini:



Gambar 3. Representasi Relasi Himpunan

Terakhir adalah batasan-batasan yang dijadikan acuan dalam pembuatan jadwal sidang tugas akhir sesuai parameter penjadwalan yang ditentukan yaitu kasus duplikasi atau bentrok dan kesesuaian kategori. Setiap pelanggaran dari batasan-batasan tersebut berdampak pada kenaikan jumlah pelanggaran atau penalti pada setiap batasan. Sebagai catatan supaya mempermudah penggambaran kasus, dideklarasikan variabel-variabel berikut, R adalah posisi ruangan, dan T adalah slot waktu. Ketika variabel menunjukkan R1T1 berarti data tersebut adalah untuk ruangan 1 dan slot waktu 1.

1. HC1: Tidak ada anggota himpunan penguji MP yang sama dalam ruangan yang sama di waktu yang bersamaan.

2. HC2: Tidak ada anggota himpunan penguji MP yang sama pada ruangan yang berbeda di waktu yang bersamaan.
3. HC3: Tidak ada anggota himpunan penguji MP yang sama dengan himpunan relasi MD yaitu mahasiswa dengan dosen pembimbing.
4. HC4: Minimal setengah dari jumlah himpunan penguji memiliki kesamaan antara himpunan MKt (kategori tugas akhir mahasiswa) dengan Dkt (spesialisasi dosen penguji).

Hasil penjadwalan yang valid adalah tidak ada batasan-batasan tersebut yang dilanggar atau nilai penalti sama dengan nol.

Kromosom

Kromosom, dalam penelitian ini adalah sebuah jadwal yang sudah terbagi oleh ruangan, hari dan waktu, dan dikodekan sedemikian rupa agar dapat diproses oleh operator-operator genetika. Alasan tersebut yang menyebabkan anggota dari himpunan dinyatakan dalam suatu bentuk kode yang mewakili nilai aslinya.

Tidak ada aturan khusus dalam penentuan pengkodean tersebut, selama pengkodean tersebut sesuai dengan batasan-batasan yang sudah ditentukan, dapat merepresentasikan solusi masalah yang dioptimasi, serta dapat diproses oleh operator genetika (Zukhri, 2014). Penulis merepresentasikan kromosom seperti Tabel 1 di bawah ini.

Tabel 1 Representasi Kromosom pada GA Penjadwalan

	Hari 1
	Waktu 1
Ruang 1	GEN
Ruang 2	GEN

Pada Tabel 1 di atas terdapat istilah gen. Gen sendiri merupakan bagian dari kromosom, atau disebut juga sebagai sub penyelesaian. Gen dalam penjadwalan ini adalah sidang yang dilaksanakan pada ruangan dan slot waktu tertentu.

Pada Gambar 3 sebelumnya menunjukkan bahwa gen adalah irisan dari himpunan MD, MP, DKt dan MKt. Sehingga gen direpresentasikan sebagai berikut:

$$Gen[R][D][T] = \{MD_n, MP_n, MKt_n, DKt_n\}$$

Nilai Kebugaran (Fitness Value)

Nilai kebugaran atau fitness value digunakan untuk memberikan nilai pada suatu kromosom, sehingga dapat ditentukan baik tidaknya sebuah kromosom terhadap batasan penjadwalan. Pada sub bagian pemodelan kasus telah dijelaskan Hasil penjadwal yang valid adalah tidak ada batasan-batasan tersebut yang dilanggar atau sama dengan nol. Setiap pelanggaran dari batasan-batasan tersebut maka akan berdampak pada kenaikan jumlah pelanggaran atau penalti untuk setiap batasan, dirumuskan pada persamaan (1) sebagai berikut:

$$HC_n = a + 1 \tag{1}$$

$$\begin{aligned} HC_n &= \text{Batasan ke-}n \\ a &= \text{nilai awal penalti} \end{aligned}$$

Berdasarkan persamaan tersebut maka, untuk menentukan seberapa baik atau *fitness value* jadwal yang ada digunakan fungsi minimasi pada persamaan (2) berikut:

$$F = \frac{1}{1 + (HC_1 + HC_2 + HC_3 + HC_4)} \tag{2}$$

$$F = \text{nilai fitness dari jadwal}$$

Fungsi minimasi (Ferawaty, 2010) digunakan karena kromosom terbaik adalah kromosom yang memiliki nilai penalti lebih sedikit. Sehingga untuk mendapatkan nilai lebih besar pada penalti yang lebih kecil digunakan fungsi minimasi. Angka satu pada denumerator digunakan agar nilai maksimal yang didapatkan adalah 1 (untuk nilai konflik = 0) dan mencegah denominator = 0.

Populasi

Pada GA, kromosom tidak hanya dibangkitkan satu saja, melainkan bervariasi tergantung nominal yang ditentukan, hal ini disebut dengan populasi. Populasi atau kumpulan dari representasi solusi atau kromosom, dimana masing-masing solusi tersebut memiliki nilai kebugaran yang cara perhitungannya telah dijelaskan pada poin sebelumnya. Solusi atau kromosom terbaik di dalam populasi tersebut adalah solusi paling optimal dari permasalahan. Representasi populasi bisa dilihat pada Gambar 4 di bawah ini.

A	
Hari 1	
Waktu 1	
Ruang 1	MD[ta1, (ds1, ds2)] MP[ta1, (ds3, ds4, ds5)] MKt[ta1, sw] DKt[(ds3, sw), (ds4, hw), (ds5, sw)]
Ruang 2	MD[ta2, (ds1, ds2)] MP[ta2, (ds6, ds6, ds8)] MKt[ta2, sw] DKt[(ds6, sw), (ds6, sw), (ds8, hw)]
HC1	1
HC2	0
HC3	0
HC4	0
Total	1
Fitness	0,5

B	
Hari 1	
Waktu 1	
Ruang 1	MD[ta1, (ds1, ds2)] MP[ta1, (ds3, ds4, ds5)] MKt[ta1, sw] DKt[(ds3, sw), (ds4, hw), (ds5, sw)]
Ruang 2	MD[ta2, (ds1, ds2)] MP[ta2, (ds4, ds6, ds8)] MKt[ta2, sw] DKt[(ds6, sw), (ds6, sw), (ds8, hw)]
HC1	0
HC2	1
HC3	0
HC4	0
Total	1
Fitness	0,5

Gambar 4. Representasi Populasi dengan dua kromosom, yaitu A dan B

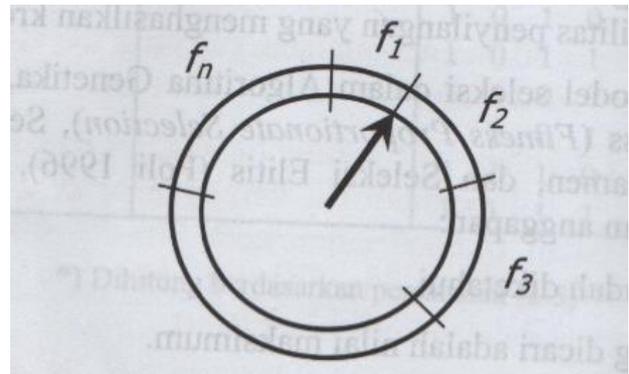
Pada penelitian ini untuk pembangkitan populasi, nilai himpunan MD (Tugas Akhir dengan dosen pembimbing), dan MKt (Tugas

Akhir dengan kategori) dibangkitkan secara acak, kemudian diberikan langsung kepada setiap kromosom pada populasi. Sehingga masing-masing kromosom telah memiliki nilai MD dan MKt yang sama, tetapi berbeda pada nilai MP (Penguji Tugas Akhir) dan DKt (Kategori Dosen Penguji).

Seleksi

Ketika populasi sudah dibangkitkan secara acak, maka seleksi kromosom atau individu dari populasi yang dijadikan sebagai induk untuk individu baru dapat dilakukan. Metode untuk mendapatkan calon induk yang digunakan pada penelitian ini dinamakan *roulette wheel* dengan seleksi peringkat.

Metode ini menggunakan penggambaran roda rolet yang terbagi atas masing-masing kromosom yang besarnya tergantung pada probabilitas yang didapatkan berdasarkan fungsi distribusi linier, sedemikian sehingga didapatkan jumlah seluruh probabilitas sama dengan satu. Berikut adalah ilustrasi dari roda rolet dan perumusan dari fungsi distribusi linier, terdapat pada gambar dan persamaan di bawah ini (Zukhri, 2014).



Gambar 5. Ilustrasi dari *Roulette Wheel*

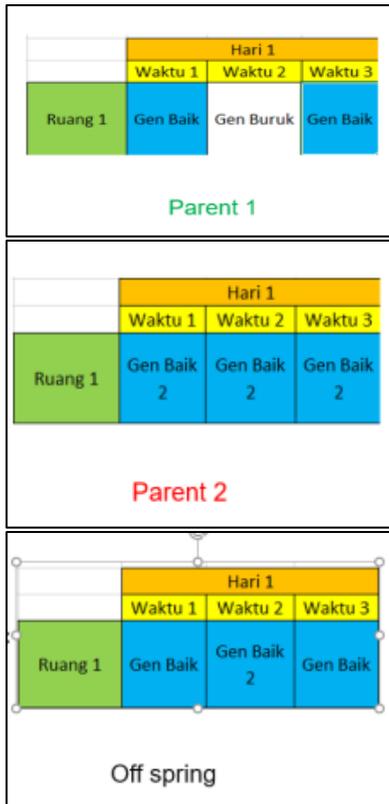
$$p_i = a \cdot i + b \text{ dengan } \sum_{i=1}^N p_i = 1 \tag{3}$$

i = indek atau deret ke - i
N = jumlah populasi

Crossover

Setelah induk dipilih pada proses seleksi dilakukan tahapan pembentukan kromosom baru salah satunya adalah *crossover* atau persilangan genetika. Persilangan genetika adalah proses pembentukan kromosom baru dengan mengabungkan dua karakter induk yang lebih baik. Metode penyilangan yang digunakan dalam penelitian ini adalah *uniform crossover* (Zukhri,

2014), dimana seluruh gen pada kromosom baru adalah cerminan dari induk pertama, dengan nilai gen yang lebih buruk dari induk tersebut digantikan dengan gen induk kedua. Ilustrasi dari proses tersebut dapat dilihat pada Gambar 6 di bawah ini.



Gambar 6. Ilustrasi Proses Persilangan Genetik (Crossover) pada GA

Mutasi

Selain persilangan genetik, pada pembentukan kromosom baru bisa juga terjadi proses mutasi. Mutasi adalah proses munculnya gen acak yang tidak dimiliki oleh kedua induk dari kromosom baru. Mutasi disini terjadi karena sebuah nilai probabilitas yang sangat kecil antara 0 – 0.3 (Robandi, 2019), sehingga proses ini tidak selalu terjadi pada operasi genetika.

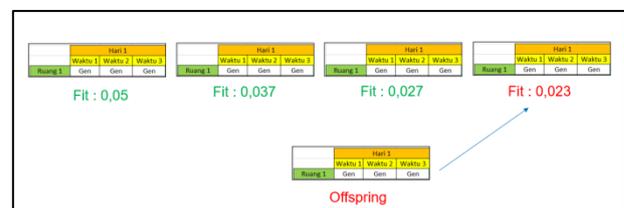
Pada penelitian ini proses mutasi yang digunakan adalah scramble mutation (Zukhri, 2014), yaitu algoritma memilih posisi gen secara acak untuk digantikan nilainya dengan gen baru yang dibangkitkan. Proses tersebut diilustrasikan pada Gambar 7 di bawah ini.



Gambar 7. Ilustrasi Proses Mutasi pada GA

Generasi

Generasi adalah proses pembentukan populasi baru yang dilakukan dengan menggantikan kromosom terburuk dari populasi sebelumnya, dengan kromosom baru yang dihasilkan dari operasi genetika. Proses tersebut diilustrasikan pada Gambar 8 di bawah ini.



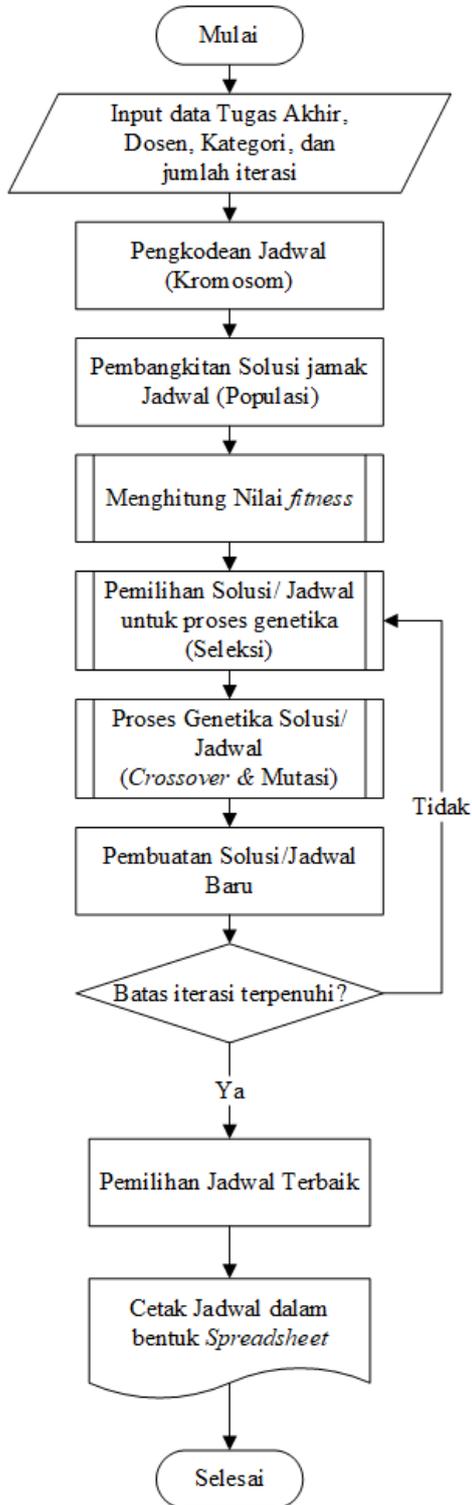
Gambar 8. Ilustrasi Pembentukan Populasi Baru

Berdasarkan Gambar 8 di atas, kromosom terburuk otomatis digantikan dengan offspring atau individu baru terlepas dari lebih baik atau tidaknya nilai fitness antara keduanya.

Desain Sistem

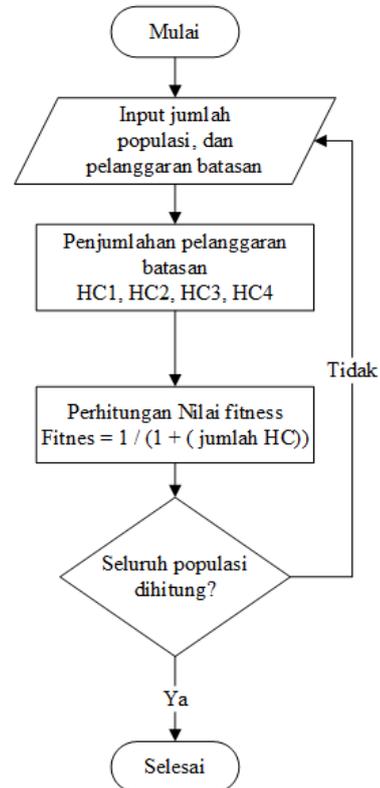
Desain sistem adalah tahapan kedua pada metodologi waterfall. Pada tahapan ini kebutuhan sistem yang sudah dijelaskan pada bagian sebelumnya dipelajari lebih lanjut untuk

mempermudah menentukan kebutuhan perangkat keras (bila ada) dan persyaratan sistem. Dalam menjelaskan system design digunakan diagram alir dari proses algoritma genetika (GA), dan penjelasan sub proses dari diagram alir tersebut, disertai dengan variabel input dan keluaran yang dihasilkan. Gambar 9 diagram alir dari proses algoritma genetika.

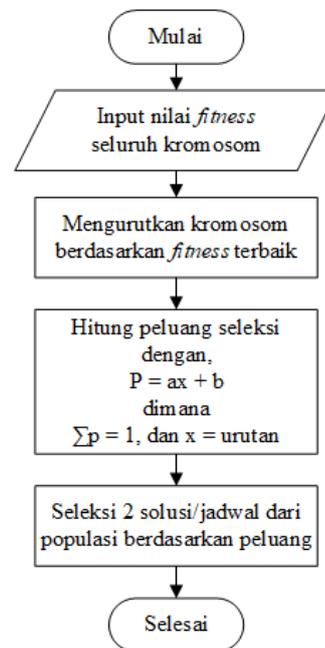


Gambar 9. Diagram Alir Seluruh Proses GA

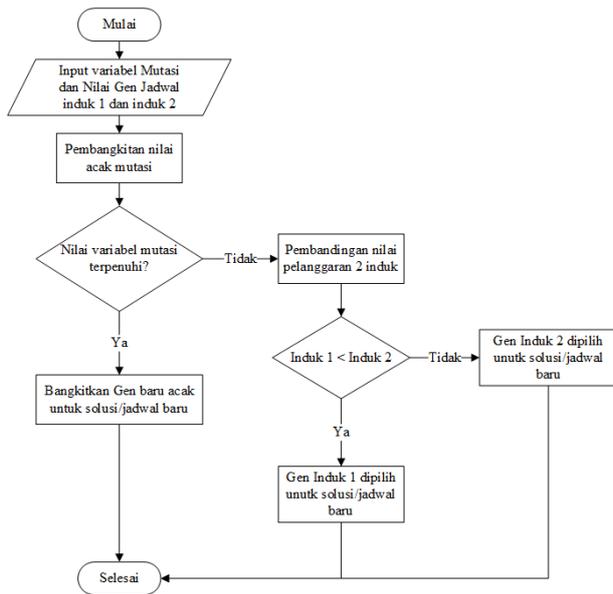
Gambar 9 di atas menunjukkan alur proses GA pada sistem yang dibuat, terdapat 3 buah sub proses pada diagram alir tersebut yang masing-masing ditunjukkan pada Gambar 10, Gambar 11, dan Gambar 12 secara berurutan.



Gambar 10. Diagram Alir Perhitungan Fitness Value Kromosom GA



Gambar 11. Diagram Alir Proses Seleksi Induk pada GA



Gambar 12. Diagram Alir Proses Crossover dan Mutasi pada GA

III. HASIL

Pengujian adalah tahap akhir dari penelitian ini dan merupakan pemaparan dari fase integration and testing pada metodologi *waterfall*. Parameter penjadwalan yang difokuskan dalam pengujian ada dua yaitu duplikasi atau bentrok dan kesesuaian kategori. Berikut poin penjabarannya:

1. Duplikasi yaitu pengujian untuk batasan HC1, HC2, dan HC3
2. Kesesuaian kategori antara tugas akhir dengan dosen penguji yaitu batasan HC4.

Terdapat empat buah skenario dalam pengujian ini dimana pada setiap skenario terdapat variabel yang diubah, yaitu variabel jumlah ruangan mulai dari 1-5 ruangan. Setiap perubahan variabel pengujian, dilakukan pengambilan data sebanyak 20 kali. Tabel 2 di bawah ini menunjukkan keterangan skenario pengujian.

Tabel 2. Skenario Pengujian Algoritma

No	Nama	Dosen Penguji	Pengujian per Ruang	Jumlah Ruang				
1	Skenario 1	23	4	1	2	3	4	5
2	Skenario 2		2	1	2	3	4	5
3	Skenario 3	15	4	1	2	3	4	5

No	Nama	Dosen Penguji	Pengujian per Ruang	Jumlah Ruang				
4	Skenario 4		2	1	2	3	4	5

Tabel 3 di bawah menunjukkan variabel dalam pengujian algoritma genetika (GA).

Tabel 3. Variabel Digunakan Dalam Pengujian Sistem GA yang Dibuat

Parameter	Nilai
Jumlah Populasi	6
Maksimum Iterasi	100000
Probabilitas Mutasi	0 - 0.3 (bergantung pada besar gen)
Pengujian per perubahan variabel	20
Jumlah Mahasiswa	55
Jumlah Waktu per Hari	3

Hasil pengujian yang ditampilkan berupa persentase keberhasilan program algoritma genetika dalam pembuatan jadwal yang valid, yaitu jadwal dengan nilai penalti sama dengan nol. Sehingga dalam 20 kali pengujian apabila seluruh jadwal yang dihasilkan valid maka persentase keberhasilannya sama dengan 100%, apabila terdapat jadwal yang tidak valid saat dilakukan pengujian maka persentase keberhasilannya akan berkurang, tergantung pada jumlah jadwal yang dapat.

Pengambilan data penalti atau pelanggaran batasan pada generasi jadwal dilakukan menggunakan aplikasi POSTMAN dengan membandingkan antara kromosom terbaik sebelum GA dan setelah GA. Contoh penalti yang dihasilkan saat generasi jadwal ditunjukkan oleh Gambar 13 di bawah ini.

```

Sebelum
{
  "hc1totb": 9,
  "hc2totb": 15,
  "hc3totb": 16,
  "hc4totb": 10,
  "fitnessb": 0.020000000000000004163336342344337026588618755340576171875
}
    
```

```
Sesudah
{
  "hc1tot": 0,
  "hc2tot": 0,
  "hc3tot": 0,
  "hc4tot": 0,
  "fitness": 1
}
```

Gambar 13. Hasil Penalti Generasi Jadwal Sebelum dan Setelah GA

Berikut adalah contoh hasil pengambilan data penalti sebanyak 20 kali untuk persentase keberhasilan generasi jadwal masing-masing pengujian, baik duplikasi maupun kategori ditunjukkan pada Gambar 14 dan Gambar 15 di bawah ini.

No	Individu Terbaik Pra GA					Individu Terbaik Pasca GA					Time (s)
	Total He1	Total He2	Total He3	Total Konflik	Fitness	Total He1	Total He2	Total He3	Total Konflik	Fitness	
1	15	25	15	55	0,018	0	0	0	0	1,000	9,14
2	10	32	13	55	0,018	0	0	0	0	1,000	12,9
3	15	34	11	60	0,016	0	0	0	0	1,000	15,43
4	14	29	16	59	0,017	0	0	0	0	1,000	5,88
5	14	25	18	57	0,017	0	1	0	1	0,500	29,44
6	13	25	17	55	0,018	0	0	0	0	1,000	7,32
7	7	28	18	53	0,019	0	0	0	0	1,000	6,58
8	8	32	18	58	0,017	0	0	0	0	1,000	11,01
9	14	17	10	41	0,024	0	0	0	0	1,000	8,13
10	14	29	16	59	0,017	0	0	0	0	1,000	10,8
11	11	32	25	68	0,014	0	0	0	0	1,000	4,82
12	15	29	22	66	0,015	0	0	0	0	1,000	28,11
13	11	22	19	52	0,019	0	1	0	1	0,500	38,65
14	11	34	14	59	0,017	0	0	0	0	1,000	33,68
15	7	30	20	57	0,017	0	0	0	0	1,000	23,64
16	12	34	14	60	0,016	0	0	0	0	1,000	11,42
17	7	39	14	60	0,016	0	0	0	0	1,000	25,43
18	14	27	24	65	0,015	0	0	0	0	1,000	6,19
19	18	26	17	61	0,016	0	0	0	0	1,000	25,43
20	18	27	16	61	0,016	0	0	0	0	1,000	10,83
Modus	14	25	16	55	0,018	0	0	0	0	1,000	
Min	7	17	10	41	0,014	0	0	0	0	0,500	
Max	18	39	25	68	0,024	0	1	0	1	1,000	
Rata2	12,4	28,8	16,9	58,1	0,017	0,0	0,1	0,0	0,1	0,950	16,24
Error				26,4					0,0		
Success Rate (%)											
90,00											
Sebelum						Setelah					

Gambar 14. Contoh Hasil Pengujian Keberhasilan untuk Kasus Duplikasi

Pada Gambar 14, persentase keberhasilan algoritma adalah 90%, karena masih terdapat 2 jadwal tidak valid dari 20 kali pengujian. Pada Gambar 15 persentase keberhasilannya berbeda yaitu sebesar 85% karena terdapat 3 jadwal yang tidak valid dari 20 kali pengujian.

No	Individu Terbaik Pra GA	Individu Terbaik Pasca GA	Time (s)
	Total Hc1	Total Hc1	
1	20	0	29,45
2	17	1	30,22
3	14	0	30,52
4	14	0	36,79
5	13	0	30,28
6	14	0	32,8
7	18	0	31,2
8	12	0	31,5
9	16	1	29,44
10	18	0	30,16
11	16	0	30,9
12	18	1	30,82
13	14	0	31,42
14	16	0	30,95
15	17	0	28,09
16	10	0	29,46
17	10	0	32,85
18	12	0	32,52
19	24	0	32,14
20	12	0	31,52
Modus	14	0	
Min	10	0	
Max	24	1	
Rata2	15,3	0,2	31,152
Success Rate (%)		85	
Sebelum		Setelah	

Gambar 15. Contoh Hasil Pengujian Keberhasilan untuk Kasus Kesesuaian Kategori

Berdasarkan contoh perhitungan pada paragraph sebelumnya, berikut adalah persentase keberhasilan dari setiap skenario pengujian yang dilakukan untuk masing-masing kasus (duplikasi dan kategori) digambarkan pada Tabel 4 dan Tabel 5 di bawah ini.

Tabel 4. Hasil Pengujian Keseluruhan Skenario untuk Kasus Duplikasi

No	Nama	Persentase Keberhasilan per Jumlah Ruang (%)				
		1	2	3	4	5
1	Skenario 1	100	100	90	5	0
2	Skenario 2	100	100	100	100	100
3	Skenario 3	85	45	0	0	0
4	Skenario 4	100	100	100	100	55
Rata-rata (%)		96,25	86,25	72,5	51,25	38,75

Berdasarkan hasil di atas, bagaimana persentase tersebut dapat berubah seiring dengan perubahan variabel yang diujikan, penulis memasukkan rasio

antara jumlah dosen penguji dengan jumlah kebutuhan dosen penguji perslot waktu sidang.

Tabel 5. Hasil Pengujian Keseluruhan Skenario untuk Kasus Kesesuaian Kategori

No	Nama	Persentase Keberhasilan per Jumlah Ruang (%)				
		1	2	3	4	5
1	Skenario 1	100	100	100	85	35
2	Skenario 2	100	100	100	100	100
3	Skenario 3	85	80	45	10	0
4	Skenario 4	100	100	100	100	100
Rata-rata (%)		96,25	95	86,25	73,75	58,75

Apabila rasio ini dibandingkan dengan data keberhasilan pada Tabel 4 dan Tabel 5, maka rasio untuk mendapatkan keberhasilan pembuatan jadwal yang optimal akan didapatkan, apabila rasio antara jumlah penguji dan jumlah kebutuhan penguji perwaktu di atas 2:1 untuk mendapatkan keberhasilan di atas 90% pada kasus duplikasi. dan 1,5:1 pada kasus kesesuaian kategori.

Tabel 6. Rasio Jumlah Penguji dengan Kebutuhan Penguji Per Satu Waktu

No	Nama	Persentase Keberhasilan per Jumlah Ruang (%)				
		1	2	3	4	5
1	Skenario 1	5,75	2,88	1,92	1,44	1,15
2	Skenario 2	11,50	5,75	3,83	2,88	2,30
3	Skenario 3	3,75	1,88	1,25	0,90	0,75
4	Skenario 4	7,50	3,75	2,50	1,88	1,50
Rata-rata (%)		7,13	3,56	2,38	1,77	1,43

IV. PENUTUP

Algoritma genetika dapat menjadi solusi untuk masalah penjadwalan sidang tugas akhir dengan cara membangkitkan populasi atau kumpulan jadwal secara acak, kemudian jadwal tersebut akan saling membandingkan dan memperbaiki satu sama lain menggunakan operasi genetika. Kemudian dipilihlah hasil jadwal yang paling optimal, yaitu tidak memiliki pelanggaran batasan.

Selain itu, algoritma genetika dapat menyesuaikan dengan perubahan variabel pada penjadwalan, hanya saja perubahan variabel tersebut akan mempengaruhi sebuah rasio yang

berdampak pada persentase keberhasilan algoritma dalam pembuatan jadwal yang valid.

V. DAFTAR PUSTAKA

- Abdullah, S., & Turabieh, H. (2008). Generating university course timetable using Genetic Algorithms and local search. *Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008, 1(November)*, 254–260. <https://doi.org/10.1109/ICCIT.2008.379>
- Bulman, M. (2017). SDLC - Waterfall Model. *The Independent*, May.
- Ferawaty, E. (2010). *Optimasi Penjadwalan Mata Kuliah pada Perguruan Tinggi dengan Menggunakan Algoritma Genetika*.
- Fiarni, C., Gunawan, A. S., Ricky, Maharani, H., & Kurniawan, H. (2015). Automated Scheduling System for Thesis and Project Presentation Using Forward Chaining Method with Dynamic Allocation Resources. *Procedia Computer Science*, 72, 209–216. <https://doi.org/10.1016/j.procs.2015.12.133>
- Mawaddah, N. K., & Mahmudy, W. F. (2006). Optimasi Penjadwalan Ujian menggunakan Algoritma Genetika. *Kursor*, 2(2), 1–8. wayanfm@ub.ac.id
- Permadi, I. (2010). Penerapan Algoritma Genetika untuk Optimasi Penjadwalan Tebangan Hutan (Applying of Genetic Algorithm for Scheduling Optimization Cuts Away Forest). *Juita*, 1, 19–27.
- Pinedo, M. L. (2016). *Scheduling Theory, Algorithm and Systems* (5th ed.). Springer. <https://doi.org/10.1007>
- Robandi, I. (2019). *Artificial Intelligence - Mengupas Rekayasa Kecerdasan Tiruan* (M. Kika (ed.); 1st ed.). ANDI.
- 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*, 6(1), 85. <https://doi.org/10.25126/jtiik.2019611262>
- Zukhri, Z. (2014). *ALGORITMA GENETIKA, Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi* (Seno (ed.); 1st ed.). ANDI.