

Perbandingan *Fluid Genetic Algorithm* dan *Genetic Algorithm* untuk Penjadwalan Perkuliahan

Muhammad Ezar Al Rivan^{[1]*}, Bhagaskara^[2]

Teknik Informatika^{[1], [2]}

STMIK Global Informatika MDP

Palembang, Indonesia

meedzhar@mdp.ac.id^[1], bhagaskaraliancer@gmail.com^[2]

Abstract— The lecture schedule is a problem that belongs to the NP-Hard problem and multi-objective problem because it has several variables that affect the preparation of the schedule and has limitations that must be met. One solution that has been found is using a Genetic Algorithm (GA). GA has been proven to be able to provide a schedule that can meet limitations in scheduling. Besides, it also found a new concept of thought from GA, namely the Fluid Genetic Algorithm (FGA). The most visible difference between FGA and GA is that there is no mutation process in each iteration. FGA has a new stage, namely individual born and new constants, namely global learning rate, individual learning rate, and diversity rate. This concept of thinking was tested in previous studies and found that FGA is superior to GA for the problem of finding the optimum value of a predetermined function, but this function is not included in the multi-objective problem. In this study, the testing and comparison of FGA and GA were conducted for the problem of scheduling lectures at STMIK XYZ. Based on the results obtained, FGA can produce a schedule without any hard constraint violations. FGA can be used to solve multi-objective problems. FGA has a smaller number of generations than GA. However, overall GA is superior in producing schedules without any problems.

Keywords— *Genetic Algorithm, Fluid Genetic Algorithm, Timetabling*

Abstrak— Penjadwalan perkuliahan merupakan masalah yang termasuk pada NP-Hard problem dan multi-objective problem dikarenakan memiliki beberapa variabel yang mempengaruhi penyusunan jadwal serta memiliki batasan-batasan yang harus dipenuhi. Salah satu solusi penyelesaian yang telah ditemukan adalah menggunakan Algoritma Genetika (GA). GA telah terbukti dapat memberikan jadwal yang dapat memenuhi batasan-batasan dalam penyusunan jadwal. Selain itu ditemukan juga konsep pemikiran baru dari GA yakni *Fluid Genetic Algorithm* (FGA). Perbedaan yang sangat terlihat dari FGA dibandingkan dengan GA adalah tidak adanya proses mutasi dalam setiap iterasinya. FGA memiliki tahap baru yakni *born individual* dan konstanta baru yakni *global learning rate*, *individual learning rate*, dan *diversity rate*. Konsep pemikiran ini diuji pada penelitian sebelumnya dan mendapati FGA lebih unggul dibandingkan dengan GA untuk masalah pencarian nilai optimum dari suatu fungsi yang telah ditentukan, tetapi fungsi tersebut tidak termasuk dalam *multi-objective problem*. Pada

penelitian ini dilakukanlah pengujian dan perbandingan FGA dan GA untuk masalah penjadwalan perkuliahan di STMIK XYZ. Berdasarkan hasil yang didapat FGA dapat menghasilkan jadwal tanpa adanya pelanggaran *hard constraint*. FGA bisa digunakan untuk penyelesaian *multi objective problem*. FGA memiliki jumlah generasi yang lebih sedikit dibandingkan GA. Namun secara keseluruhan GA lebih unggul dalam menghasilkan jadwal tanpa ada masalah.

Kata Kunci— *Genetic Algorithm, Fluid Genetic Algorithm, Penjadwalan*

I. INTRODUCTION

Penjadwalan perkuliahan merupakan masalah yang termasuk *Non Polynomial (NP)-hard problem* yakni masalah yang mempunyai kompleksitas waktu penyelesaian dengan tingkatan polinomial [1] [2] [3]. Hal ini terjadi dikarenakan dalam masalah penjadwalan perkuliahan terdapat beberapa variabel yang menjadi pengaruh yakni ruangan, kelas, waktu, mata kuliah, dan dosen. Variabel-variabel tersebut yang akan disusun dalam satu tabel jadwal dan ditentukan seberapa bagus hasilnya berdasarkan suatu fungsi objektif.

Pada pemasalahan penjadwalan perkuliahan terdapat dua fungsi objektif yakni fungsi yang dipengaruhi oleh *hard constraints* dan fungsi yang dipengaruhi oleh *soft constraints* [1], [4]. Jenis permasalahan yang memiliki lebih dari satu fungsi objektif seperti ini termasuk pada *multi-objective problem* [5] [6]. Dalam penyelesaian masalah atau penghasilan solusi, algoritma yang digunakan haruslah menghasilkan nilai yang terbaik pada kedua fungsi tersebut sehingga menghasilkan solusi yang terbaik. Salah satu algoritma yang digunakan dalam menyelesaikan masalah ini adalah *genetic algorithm* [7]. Penelitian lain yang menggunakan *genetic algorithm* yaitu pada penelitian [8] untuk menjadwalkan penggunaan sumber daya seefisien mungkin. Penelitian tersebut menerapkan *genetic algorithm* sebagai penjawal tugas pada *server*. Penelitian [9] menggunakan *genetic algorithm* sebagai penjadwal untuk meminimalkan penggunaan energi pada *server*.

Genetic algorithm (GA) merupakan *Evolutionary Algorithm* yakni algoritma dengan metode heuristik yang dikembangkan berdasarkan konsep evolusi dalam biologi [10]. Algoritma ini memiliki tahap awal yakni *chromosome*

encoding dan population initialization, setelah itu terdapat beberapa langkah yang diulang beberapa kali yakni fitness evaluation, selection, crossover, dan mutation [11]. Pada tahap selection, individu atau kromosom yang terbaik akan terpilih dan dilakukan crossover (proses penyilangan pasangan individu) dengan individu atau kromosom terbaik lainnya dengan harapan menghasilkan offspring yang lebih baik sehingga pada akhirnya didapatkan hasil yang terbaik. Konsep GA ini terus berkembang dan ditemukanlah pemikiran baru yakni Fluid Genetic Algorithm.

Fluid Genetic Algorithm (FGA) merupakan algoritma yang memiliki konsep dan proses yang mirip dengan GA. Perbedaannya terletak pada cara pandang kromosom dan individu, representasi kromosom dan proses mutasi [12]. Pada penelitian sebelumnya FGA telah diterapkan untuk masalah sederhana yang bukan merupakan multi-objective problem yakni mencari titik optimal dari suatu fungsi. Hasil dari penelitian tersebut menunjukkan bahwa FGA lebih unggul dari GA baik dalam tingkat keberhasilan, waktu eksekusi, dan jawaban yang dihasilkan [12].

Berdasarkan uraian tersebut, GA telah berhasil diterapkan untuk masalah multi-objective problem seperti penjadwalan perkuliahan sedangkan FGA belum diterapkan sehingga belum diketahui hasil performanya. Maka dilakukan penelitian dengan permasalahan bagaimana perbandingan performa antara FGA dan GA untuk masalah penjadwalan perkuliahan. Penjadwalan perkuliahan sebelumnya menggunakan GA sehingga pada penelitian ini FGA digunakan untuk melakukan penjadwalan kuliah sehingga dapat diketahui performanya dibandingkan dengan GA

II. TINJAUAN PUSTAKA

GA digunakan untuk penjadwalan seperti pada penelitian yang dilakukan [4] dan [11]. Pada penelitian tersebut dihasilkan jadwal yang optimal. GA tidak hanya digunakan untuk penjadwalan perkuliahan saja namun dapat digunakan pada penjadwalan penggunaan sumber daya [8] dan [9] sehingga penggunaan sumber daya menjadi minimum.

GA mengalami perkembangan untuk meningkatkan performanya. Pada penelitian yang dilakukan oleh [12] digunakan FGA yang memodifikasi GA dengan cara menghilangkan proses dalam GA yaitu mutation. Pada penelitian tersebut diketahui FGA lebih unggul dibandingkan dengan GA. Pada penelitian tersebut FGA digunakan untuk melakukan optimasi fungsi matematika.

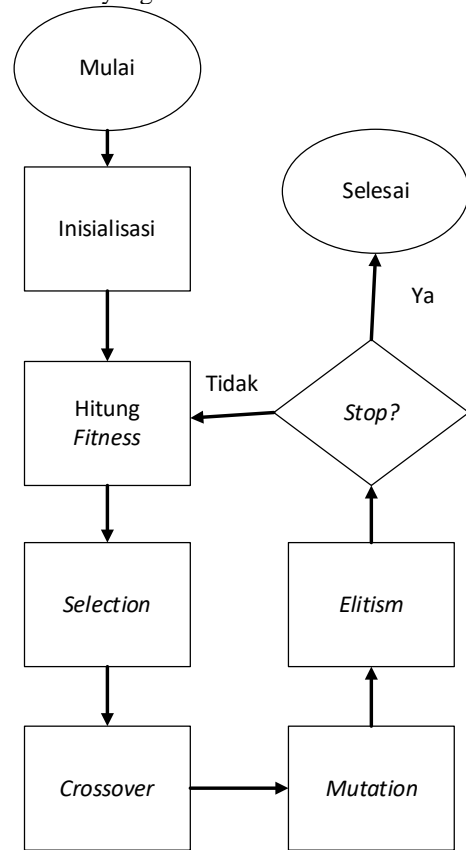
III. METODE PENELITIAN

Implementasi algoritma dilakukan dengan bahasa pemrograman Java dan IntelliJ IDEA 2017.1.4 sebagai Integrated Development Environment (IDE) dengan rancangan sebagai berikut:

A. Genetic Algorithm (GA)

Penerapan algoritma GA memiliki 7 tahapan. Langkah-langkah dapat dilihat pada Gambar 1 dan berikut penjelasan setiap tahapannya:

- 1) Inisialisasi merupakan tahap dilakukannya pembentukan individu-individu (populasi) awal dengan representasi matriks dua dimensi yang berisi nilai acak sebanyak jumlah individu yang telah ditentukan



Gambar 1. Flowchart GA [12]

- 2) Hitung fitness dengan menggunakan Persamaan 1 yang merupakan tahap dilakukannya perhitungan nilai objektif dari setiap individu yang menggunakan fungsi sebagai berikut [13]

$$f(HC, SC) = \frac{1}{1 + (\sum HC * 10) + (\sum SC * 3)} \tag{1}$$

Keterangan :

HC = Hard Constraint

SC = Soft Constraint

- 3) Selection merupakan tahap dipilihnya individu-individu untuk generasi selanjutnya dengan metode roulette wheel yang banyaknya individu terpilih bergantung pada crossover rate [12].
- 4) Crossover merupakan tahap persilangan antara dua individu yang telah dipilih dengan menggunakan metode two-dimensional graphical crossover [14].
- 5) Mutation merupakan tahap dilakukannya perubahan nilai sel dari individu hasil crossover yang banyaknya bergantung pada mutation rate sedangkan banyaknya pilihan individu yang dapat terpengaruh bergantung pada mutation [14].

- 6) *Elitism* merupakan proses untuk mempertahankan individu-individu terbaik yang pernah terbentuk di suatu generasi. Salah satu cara *elitism* pada umumnya adalah membuat salinan individu-individu terbaik pada generasi sekarang untuk dibawa ke generasi selanjutnya [15]. Maka pada tahap ini dilakukan pemilihan individu terbaik diluar hasil *crossover* untuk dibawa ke generasi selanjutnya yang banyaknya bergantung pada selisih jumlah individu yang telah ditentukan di awal dengan jumlah individu yang dihasilkan pada tahap *crossover*.
- 7) Kriteria *Stop* merupakan kondisi ketika proses iterasi berhenti yakni pada iterasi yang ke-1000.

B. Fluid Genetic Algorithm (FGA)

FGA merupakan konsep baru yang merupakan pengembangan dari algoritma GA [12]. Penerapan algoritma FGA memiliki 8 tahapan. Langkah-langkah dapat dilihat pada Gambar 2 dan berikut penjelasan setiap tahapannya:

1. Inisialisasi merupakan tahap dilakukannya pembentukan kromosom-kromosom awal dengan representasi matriks dua dimensi yang memiliki nilai probabilitas sebesar 0.5 sebanyak jumlah kromosom yang telah ditentukan.
2. Hitung *Blueprint* merupakan tahap dilakukannya perhitungan nilai rata-rata probabilitas seluruh kromosom dari setiap generasinya.
3. Cetak Individu merupakan tahap dilakukannya pembentukan individu untuk setiap kromosomnya yang banyaknya bergantung pada konstanta yang telah ditentukan.
4. Hitung *fitness* merupakan tahap dilakukannya perhitungan nilai objektif dari setiap individu dengan menggunakan Persamaan 1 yakni fungsi yang sama dengan GA.
5. *Selection* merupakan tahap dipilihnya individu-individu dengan metode yang sama dengan GA.
6. *Crossover* merupakan tahap persilangan antara dua individu yang telah dipilih dengan menggunakan metode yang sama dengan GA tetapi menghasilkan tepat satu kromosom baru untuk dibawa ke generasi selanjutnya.
7. *Elitism* merupakan tahap dilakukannya pemilihan individu terbaik untuk dibawa ke generasi selanjutnya dengan metode yang sama dengan GA.
8. Kriteria *Stop* merupakan kondisi ketika proses iterasi berhenti yakni pada iterasi yang ke-1000.

C. Hard Constraint

Batasan-batasan pembentukan jadwal yang wajib dipenuhi yaitu:

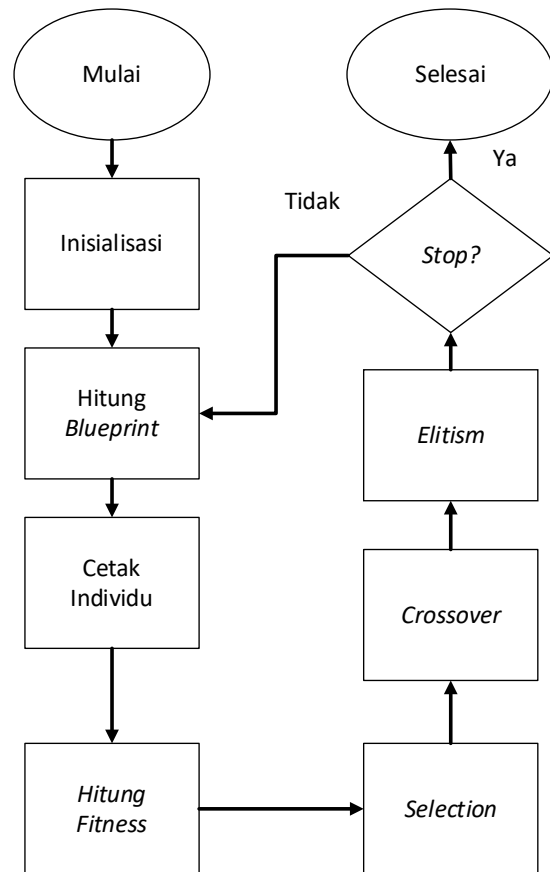
- Dosen tidak boleh mengajar di kelas yang berbeda pada satu waktu.
- Tidak ada dua atau lebih dosen di satu tempat dan waktu yang sama.

- Tidak ada dua atau lebih kelas di satu tempat dan waktu yang sama.
- Tidak ada dua atau lebih mata kuliah untuk suatu kelas di waktu yang sama.
- Seluruh mata kuliah untuk setiap kelas harus terjadwalkan.

D. Soft Constraint

Batasan-batasan pembentukan jadwal yang tidak wajib dipenuhi yaitu:

- Dosen tidak boleh mengajar tiga kali beruntun dalam satu hari.
- Dosen maksimal mengajar sebanyak 14 sks.
- Tidak ada mata kuliah di periode waktu terakhir.



Gambar 2. Flowchart FGA [12]

E. Parameter Tuning

GA dan FGA memiliki beberapa konstanta yang harus disesuaikan berdasarkan permasalahan dan pada tahap ini dilakukan pemilihan nilai konstanta GA dan FGA yang terbaik untuk permasalahan penjadwalan. Pemilihan ini dilakukan dengan cara menjalankan algoritma untuk menyusun jadwal dengan beberapa skenario. Berikut rancangan tuning parameter:

- 1) Data Jadwal

Jadwal yang digunakan untuk parameter tuning adalah jadwal yang diperoleh dari STMIK XYZ untuk program studi ABC semester Ganjil dan Genap

2) Parameter Tuning FGA

Parameter FGA yang disesuaikan adalah *Crossover Rate* (COR), *Individual Learning Rate* (ILR), *Global Learning Rate* (GLR), dan *Diversity Rate* (DR). Terdapat tiga skenario *parameter tuning* untuk FGA yang masing-masing dijalankan sebanyak 20 kali eksekusi yakni:

- Skenario dengan 10 kromosom dan 2 individu setiap kromosom.
- Skenario dengan 20 kromosom dan 1 individu setiap kromosom.
- Skenario dengan 100 kromosom dan 1 individu setiap kromosom.

3) Parameter Tuning GA

Parameter GA yang disesuaikan adalah *Crossover Rate* (COR), *Mutation* (MUT), dan *Mutation Rate* (MUTR). Terdapat dua skenario *parameter tuning* untuk GA yang masing-masing dijalankan sebanyak 20 kali eksekusi yakni:

- Skenario dengan 20 individu.
- Skenario dengan 100 individu

4) Output Parameter Tuning

Parameter tuning dilakukan dengan melakukan 20 kali eksekusi dan hasilnya dicatat ke dalam tabel yang memiliki parameter:

- Generasi Jadwal Terbaik merupakan iterasi atau generasi saat terbentuknya jadwal terbaik pada suatu eksekusi. Pada hasil ini terdapat tiga hal yang dicatat yakni minimum, rata-rata, dan maksimum nilai generasi jadwal terbaik dari 20 kali eksekusi.
- Pelanggaran HC Jadwal Terbaik merupakan banyaknya pelanggaran *hard constraints* yang terdapat pada jadwal terbaik dari suatu eksekusi. Pada hasil ini terdapat tiga hal yang dicatat yakni minimum, rata-rata, dan maksimum nilai pelanggaran HC jadwal terbaik dari 20 kali eksekusi.
- Pelanggaran SC Jadwal Terbaik merupakan banyaknya pelanggaran *soft constraints* yang terdapat pada jadwal terbaik dari suatu eksekusi. Pada hasil ini terdapat tiga hal yang dicatat yakni minimum, rata-rata, dan maksimum nilai pelanggaran SC jadwal terbaik dari 20 kali eksekusi
- *Fitness* Jadwal Terbaik merupakan nilai fungsi objektif dari jadwal terbaik dari suatu eksekusi. Pada hasil ini terdapat tiga hal yang dicatat yakni minimum, rata-rata, dan maksimum nilai *fitness* jadwal terbaik dari 20 kali eksekusi

5) Penentuan Konstanta Terbaik

Pemilihan konstanta terbaik ini dipilih dengan tingkatan prioritas sebagai berikut:

- Memilih kombinasi konstanta dengan minimum pelanggaran HC jadwal terbaik yang terkecil.
- Memilih kombinasi konstanta dengan rata-rata pelanggaran HC jadwal terbaik yang terkecil.

- Memilih kombinasi konstanta dengan minimum pelanggaran SC jadwal terbaik yang terkecil.
- Memilih kombinasi konstanta dengan rata-rata pelanggaran SC jadwal terbaik yang terkecil.

F. Pengujian Algoritma

Pada tahap ini dilakukan pengujian dengan menjalankan algoritma GA dan FGA dengan menggunakan kombinasi hasil *parameter tuning*. Berikut rancangan pengujian algoritma:

1) Data Jadwal

Jadwal yang digunakan adalah jadwal dari STMIK XYZ untuk Program Studi ABC Semester Ganjil dan Genap.

2) Pengujian FGA dan GA

Terdapat empat skenario pengujian untuk FGA dan GA yang masing

-masing dijalankan sebanyak 50 kali eksekusi yakni:

- Skenario dengan 20 individu untuk semester Ganjil.
- Skenario dengan 20 individu untuk semester Genap.
- Skenario dengan 100 individu untuk semester Ganjil.
- Skenario dengan 100 individu untuk semester Genap.

Pemilihan individu 20 dan 100 ditentukan sendiri karena tidak ditetapkan aturan individu minimum dan maksimum. Pada skenario dengan 20 individu untuk algoritma FGA terlebih dahulu dilakukan pemilihan skenario terbaik antara 10 kromosom dengan 2 individu setiap kromosom atau 20 kromosom dengan 1 individu setiap kromosom karena keduanya memiliki jumlah individu sebanyak 20.

IV. HASIL DAN PEMBAHASAN

A. Hasil Parameter Tuning

1) Hasil Parameter Tuning FGA

Berdasarkan hasil *Parameter Tuning* maka didapatkan kombinasi konstanta FGA terbaik untuk setiap skenario yakni:

- a) Konstanta terbaik untuk skenario 10 kromosom dan 2 individu setiap kromosom adalah COR 0.5, ILR 0.15, GLR 0.05, dan DR 0.05 yang hasilnya dapat dilihat pada Tabel I. Hasil yang ditampilkan pada Tabel I merupakan 5 dari 81 kombinasi konstanta yang dicoba dalam tahap *Parameter Tuning*.
- b) Konstanta terbaik untuk skenario 20 kromosom dan 1 individu setiap kromosom adalah COR 0.5, ILR 0.15, GLR 0.1, dan DR 0.05 yang hasilnya dapat dilihat pada Tabel II. Hasil yang ditampilkan pada Tabel II merupakan 5 dari 81 kombinasi konstanta yang dicoba dalam tahap *Parameter Tuning*.
- c) Konstanta terbaik untuk skenario 100 kromosom dan 1 individu setiap kromosom adalah COR 0.5, ILR 0.15, GLR 0.05, dan DR 0.05 yang hasilnya dapat dilihat pada Tabel III. Hasil yang ditampilkan pada Tabel III merupakan 5 dari 81 kombinasi konstanta yang dicoba dalam tahap *Parameter Tuning*

TABLE I. HASIL PARAMETER TUNING FGA SKENARIO 10 KROMOSOM DAN 2 INDIVIDU SETIAP KROMOSOM

No	COR	ILR	GLR	DR	Generasi Jadwal Terbaik			Pelanggaran HC Jadwal Terbaik			Pelanggaran SC Jadwal Terbaik			Fitness Jadwal Terbaik		
					Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max
1	0,5	0,05	0,05	0	7	60,7	165	12	16,95	20	15	20,4	25	1	0,5	0,05
2	0,5	0,05	0,05	0,05	35	356,2	955	9	13,3	17	15	21	26	2	0,5	0,05
3	0,5	0,15	0,05	0,05	330	833,6	985	4	7,55	10	11	17,85	23	3	0,5	0,15
4	0,7	0,05	0,05	0	9	33,9	73	12	15,95	20	13	20,95	28	4	0,7	0,05
5	0,7	0,05	0,05	0,05	45	536,95	995	10	14,75	20	10	20,95	27	5	0,7	0,05

TABLE II. HASIL PARAMETER TUNING FGA SKENARIO 20 KROMOSOM DAN 1 INDIVIDU SETIAP KROMOSOM

No	COR	ILR	GLR	DR	Generasi Jadwal Terbaik			Pelanggaran HC Jadwal Terbaik			Pelanggaran SC Jadwal Terbaik			Fitness Jadwal Terbaik		
					Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max
1	0,5	0,05	0,15	0	22	83,35	143	8	13,9	18	14	20,45	27	0,0041	0,00511	0,00735
2	0,5	0,05	0,15	0,05	76	572,9	982	7	11,95	16	14	19,8	27	0,00446	0,00562	0,00704
3	0,5	0,15	0,1	0,05	335	697,2	985	3	5,85	8	12	16,95	21	0,00746	0,00915	0,01136
4	0,7	0,05	0,15	0	4	69,25	109	8	13,3	17	14	19,6	23	0,00427	0,0053	0,00709
5	0,7	0,05	0,15	0,05	128	465,65	949	8	12,15	16	13	19	24	0,00455	0,00565	0,00725

TABLE III. HASIL PARAMETER TUNING FGA SKENARIO 100 KROMOSOM DAN 1 INDIVIDU SETIAP KROMOSOM

No	COR	ILR	GLR	DR	Generasi Jadwal Terbaik			Pelanggaran HC Jadwal Terbaik			Pelanggaran SC Jadwal Terbaik			Fitness Jadwal Terbaik		
					Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max
1	0,5	0,05	0,05	0	95	115,95	138	1	4	7	10	15,55	20	0,0084	0,01193	0,01667
2	0,5	0,05	0,05	0,05	253	512,65	951	2	3,5	5	12	15,45	21	0,0098	0,01241	0,01754
3	0,5	0,15	0,05	0,05	252	761,2	995	0	2,1	3	7	12,1	18	0,01333	0,0175	0,025
4	0,7	0,05	0,05	0	79	99,05	126	1	3,35	6	12	15,95	22	0,00901	0,01252	0,01695
5	0,7	0,05	0,05	0,05	232	678,15	987	1	3,35	6	11	15,2	19	0,00943	0,01275	0,01786

Pada tahap pengujian, skenario yang digunakan hanya skenario 20 individu dan 100 individu, sehingga perlu dilakukan pemilihan antara skenario 10 kromosom dengan 2 individu setiap kromosom dan 20 kromosom dengan 1 individu setiap kromosom. Perbandingan hasil antara skenario 10 kromosom dengan 2 individu setiap kromosom dan skenario 20 kromosom dengan 1 individu setiap kromosom dapat dilihat pada Tabel XIII.

Berdasarkan Tabel IV dapat disimpulkan bahwa skenario 20 kromosom dengan 1 individu setiap kromosom lebih baik dibandingkan skenario 10 kromosom dengan 2 individu setiap kromosom. Jadi didapatkan kombinasi konstanta terbaik untuk digunakan pada proses pengujian yakni:

- a) COR 0.5, ILR 0.15, GLR 0.1, dan DR 0.05 dengan 20 kromosom dan 1 individu setiap kromosom untuk skenario pengujian 20 individu.
- b) COR 0.5, ILR 0.15, GLR 0.05, dan DR 0.05 dengan 100 kromosom dan 1 individu setiap kromosom untuk skenario pengujian 100 individu.

TABLE IV. PERBANDINGAN HASIL PARAMETER TUNING FGA

Parameter / Skenario	10 Kromosom 2 Individu setiap Kromosom	20 Kromosom 1 Individu setiap Kromosom
Minimum Generasi Jadwal Terbaik	330	335
Rata-Rata Generasi Jadwal Terbaik	833,6	697,2
Maksimum Generasi Jadwal Terbaik	985	985
Minimum Pelanggaran HC Jadwal Terbaik	4	3
Rata-Rata Pelanggaran HC Jadwal Terbaik	7,55	5,85
Maksimum Pelanggaran HC Jadwal Terbaik	10	8
Minimum Pelanggaran SC Jadwal Terbaik	11	12
Rata-Rata Pelanggaran SC Jadwal Terbaik	17,85	16,95
Maksimum Pelanggaran SC Jadwal Terbaik	23	21

Berdasarkan Tabel IV dapat disimpulkan bahwa skenario 20 kromosom dengan 1 individu setiap kromosom lebih

- baik dibandingkan skenario 10 kromosom dengan 2 individu setiap kromosom. Jadi didapatkan kombinasi konstanta terbaik untuk digunakan pada proses pengujian yakni:
- c) COR 0.5, ILR 0.15, GLR 0.1, dan DR 0.05 dengan 20 kromosom dan 1 individu setiap kromosom untuk skenario pengujian 20 individu.
 - d) COR 0.5, ILR 0.15, GLR 0.05, dan DR 0.05 dengan 100 kromosom dan 1 individu setiap kromosom untuk skenario pengujian 100 individu.

2) Hasil Parameter Tuning GA

Berdasarkan hasil *parameter tuning* maka didapatkan kombinasi konstanta yang digunakan untuk proses pengujian yakni:

- a) COR 0.7, MUT 0.3, dan MUTR 0.01 untuk skenario pengujian 20 individu yang hasilnya dapat dilihat pada Tabel V. Hasil yang ditampilkan pada Tabel V merupakan 5 dari 36 kombinasi konstanta yang dicoba dalam tahap *Parameter Tuning*.
- b) COR 0.8, MUT 0.3, dan MUTR 0.1 untuk skenario pengujian 100 individu yang hasilnya dapat dilihat pada Tabel VI. Hasil yang ditampilkan pada Tabel VI merupakan 5 dari 36 kombinasi konstanta yang dicoba dalam tahap *Parameter Tuning*.

TABLE V. HASIL PARAMETER TUNING GA SKENARIO 20 INDIVIDU

No	COR	MUT	MUTR	Generasi Jadwal Terbaik			Pelanggaran HC Jadwal Terbaik			Pelanggaran SC Jadwal Terbaik			Fitness Jadwal Terbaik		
				Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max
1	0,5	0,1	0,1	410	804,1	985	3	5,85	9	11	17,9	22	0,00704	0,00904	0,01163
2	0,7	0,1	0,01	395	816	997	0	0,75	4	11	13	17	0,01299	0,02198	0,02941
3	0,7	0,1	0,05	612	815,25	968	1	2,6	4	9	16,6	21	0,00962	0,01337	0,02083
4	0,7	0,3	0,01	648	875	995	0	0,05	1	2	6,05	10	0,03226	0,05882	0,14286
5	0,8	0,1	0,01	516	833,25	999	0	0,1	1	5	8,25	12	0,02273	0,03985	0,0625

TABLE VI. HASIL PARAMETER TUNING GA SKENARIO 100 INDIVIDU

No	COR	MUT	MUTR	Generasi Jadwal Terbaik			Pelanggaran HC Jadwal Terbaik			Pelanggaran SC Jadwal Terbaik			Fitness Jadwal Terbaik		
				Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max	Min	Rata-rata	Max
1	0,7	0,1	0,05	378	760,05	982	0	0,05	1	4	6,25	9	0,03125	0,0518	0,07692
2	0,7	0,1	0,01	360	746,25	995	0	0,05	1	4	7,75	12	0,02703	0,0417	0,05263
3	0,8	0,1	0,01	380	747,05	973	0	0,1	1	2	6,65	10	0,02439	0,0522	0,14286
4	0,8	0,3	0,05	367	746,35	996	0	0	0	0	3,4	5	0,0625	0,1473	1
5	0,8	0,3	0,1	368	728,35	946	0	0	0	0	2,55	5	0,0625	0,1687	1

B. Hasil Pengujian Algoritma

Hasil pengujian untuk setiap skenario baik FGA dan GA dicatat dan dibandingkan ke dalam suatu tabel. Tabel VII merupakan perbandingan antara FGA dan GA dengan menggunakan 20 individu dan 100 individu Berdasarkan

Tabel VII, untuk 20 individu FGA belum berhasil membentuk jadwal terbaik yang tidak memiliki pelanggaran HC tetapi pada 100 individu terlihat bahwa FGA dapat menghasilkan jadwal terbaik yang tidak memiliki pelanggaran HC.

TABLE VII. PERBANDINGAN FGA DAN GA SECARA KESELURUHAN

Parameter / Skenario	20 Individu				100 Individu			
	Ganjil		Genap		Ganjil		Genap	
	FGA	GA	FGA	GA	FGA	GA	FGA	GA
Minimum Generasi Jadwal Terbaik	161	362	214	436	382	331	340	160
Rata-Rata Generasi Jadwal Terbaik	824,06	827,62	733,36	812,78	790,04	739,4	749,58	638,3
Maksimum Generasi Jadwal Terbaik	996	997	988	997	1000	978	999	997
Minimum Pelanggaran HC Jadwal Terbaik	3	0	0	0	0	0	0	0
Rata-Rata Pelanggaran	5,8	0,08	1,8	0,02	1,96	0	0,26	0

Parameter / Skenario	20 Individu				100 Individu			
	Ganjil		Genap		Ganjil		Genap	
	FGA	GA	FGA	GA	FGA	GA	FGA	GA
HC Jadwal Terbaik								
Maksimum Pelanggaran HC Jadwal Terbaik	9	1	3	1	3	0	1	0
Minimum Pelanggaran SC Jadwal Terbaik	11	3	9	1	7	1	3	0
Rata-Rata Pelanggaran SC Jadwal Terbaik	17,08	6,74	12,96	3,08	12,84	2,38	7,52	0,54
Maksimum Pelanggaran SC Jadwal Terbaik	22	12	17	6	18	5	12	2

Berdasarkan Tabel VII dapat dilihat bahwa GA yang menggunakan 100 individu memiliki 0 pelanggaran HC yang berarti jadwal yang dihasilkan tidak memiliki tabrakan jadwal. Berbeda dengan GA, FGA memiliki pelanggaran HC 3 yang terjadi pada semester Ganjil. Walau demikian dapat dilihat ada saatnya FGA dapat menghasilkan jadwal tanpa pelanggaran HC terlihat dari minimum pelanggaran HC FGA yaitu 0.

Jika dilihat berdasarkan generasi yang dicapai oleh masing-masing algoritma, FGA mengungguli GA. FGA memiliki generasi yang lebih sedikit dibandingkan GA. Hal ini terjadi ketika individu yang digunakan 20. Berbanding terbalik saat individu yang digunakan 100, GA lebih baik dibandingkan FGA berdasarkan generasi yang dicapai.

V. KESIMPULAN

Berdasarkan analisis performa FGA dan GA untuk penjadwalan perkuliahan dapat disimpulkan FGA dapat menghasilkan jadwal yang tidak melanggar *hard constraints*. FGA dapat memberikan solusi untuk *multi-objective problem* seperti masalah penjadwalan perkuliahan. Secara keseluruhan dari semua parameter pengujian algoritma GA lebih unggul dibandingkan algoritma FGA untuk masalah penyusunan penjadwalan perkuliahan. Pada penelitian ini masih memiliki kekurangan dalam hal melakukan *Parameter Tuning*. Jika mendapatkan parameter yang sesuai diharapkan dapat meningkatkan performa FGA.

REFERENCES

[1] E. Ghasemi, P. Moradi, and M. Fathi, "Integrating ABC with genetic grouping for university course timetabling problem," in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2015, pp. 24–29, doi: 10.1109/ICCKE.2015.7365857.

[2] E. A. Abdelhalim and G. A. [El Khayat], "A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA)," *Alexandria Eng. J.*, vol. 55, no. 2, pp. 1395–1409, 2016, doi: <https://doi.org/10.1016/j.aej.2016.02.017>.

[3] R. Lewis and J. Thompson, "Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem," *Eur. J. Oper. Res.*, vol. 240, no. 3, pp. 637–648, 2015, doi: <https://doi.org/10.1016/j.ejor.2014.07.041>.

[4] A. A. Mahiba and C. A. D. Durai, "Genetic Algorithm with Search Bank Strategies for University Course Timetabling Problem," *Procedia Eng.*, vol. 38, pp. 253–263, 2012, doi: <https://doi.org/10.1016/j.proeng.2012.06.033>.

[5] Y. Lei, J. Shi, and Z. Yan, "A memetic algorithm based on MOEA/D for the examination timetabling problem," *Soft Comput.*, vol. 22, no. 5, pp. 1511–1523, 2018, doi: 10.1007/s00500-017-2886-y.

[6] L. L. [de Oliveira], A. A. Freitas, and R. Tinós, "Multi-objective genetic algorithms in the study of the genetic code's adaptability," *Inf. Sci. (Ny)*, vol. 425, pp. 48–61, 2018, doi: <https://doi.org/10.1016/j.ins.2017.10.022>.

[7] C. Akkan and A. Gülcü, "A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem," *Comput. Oper. Res.*, vol. 90, pp. 22–32, 2018, doi: <https://doi.org/10.1016/j.cor.2017.09.007>.

[8] Y. Fang, X. Xiao, and J. Ge, "Cloud computing task scheduling algorithm based on improved genetic algorithm," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 852–856, 2019, doi: 10.1109/ITNEC.2019.8728996.

[9] P. Naithani, "Genetic algorithm based scheduling to reduce energy consumption in cloud," *PDGC 2018 - 2018 5th Int. Conf. Parallel, Distrib. Grid Comput.*, pp. 616–620, 2018, doi: 10.1109/PDGC.2018.8745801.

[10] A. H. Yousef, C. Salama, M. Y. Jad, T. El-Gafy, M. Matar, and S. S. Habashi, "A GPU based genetic algorithm solution for the timetabling problem," in *2016 11th International Conference on Computer Engineering Systems (ICCES)*, 2016, pp. 103–109, doi: 10.1109/ICCES.2016.7821982.

[11] S. Parera, H. T. Sukmana, and L. K. Wardhani, "Application of genetic algorithm for class scheduling (Case study: Faculty of science and technology UIN Jakarta)," in *2016 4th International Conference on Cyber and IT Service Management*, 2016, pp. 1–5, doi: 10.1109/CITSM.2016.7577525.

[12] R. Jafari-Marandi and B. K. Smith, "Fluid Genetic Algorithm (FGA)," *J. Comput. Des. Eng.*, vol. 4, no. 2, pp. 158–167, Apr. 2017, doi: 10.1016/j.jcde.2017.03.001.

[13] R. E. Febrita and W. F. Mahmudy, "Modified genetic algorithm for high school time-table scheduling with fuzzy time window," in *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2017, pp. 88–92, doi: 10.1109/SIET.2017.8304115.

[14] W. Liu, H. Zhu, Y. Wang, S. Zhou, Y. Bai, and C. Zhao, "Topology optimization of support structure of telescope skin based on bit-matrix representation NSGA-II," *Chinese J. Aeronaut.*, vol. 26, no. 6, pp. 1422–1429, Dec. 2013, doi: 10.1016/j.cja.2013.07.046.

[15] S. Gao and C. W. de Silva, "A modified estimation distribution algorithm based on extreme elitism," *Biosystems*, vol. 150, pp. 149–166, Dec. 2016, doi: 10.1016/j.biosystems.2016.10.001.