

OPTIMASI PENJADWALAN MATA KULIAH MENGGUNAKAN METODE ALGORITMA GENETIKA DENGAN TEKNIK *TOURNAMENT SELECTION*

Yuslena Sari¹, Muhammad Alkaff², Eka Setya Wijaya³, Syarifah Soraya⁴, Dany Primanita Kartikasari⁵

¹²³⁴Program Studi Teknologi Informasi, Universitas Lambung Mangkurat

⁵Program Studi Teknologi Informasi, Universitas Brawijaya

Email: ¹yuzlena@ulm.ac.id, ²m.alkaff@ulm.ac.id, ³ekasw@ulm.ac.id, ⁴syarifahsoraya21@gmail.com,

⁵dany.jalin@ub.ac.id

(Naskah masuk: 18 November 2018, diterima untuk diterbitkan: 18 Desember 2018)

Abstrak

Bagi sebuah perguruan tinggi, penjadwalan perkuliahan merupakan suatu kegiatan yang sangat penting untuk dapat terlaksananya proses belajar mengajar yang baik. Dimana dalam proses belajar mengajar dapat dilakukan oleh semua pihak yang terkait, bukan hanya bagi dosen yang mengajar, tetapi juga bagi mahasiswa yang mengambil mata kuliah. Dalam penyusunan jadwal, ada beberapa variabel yang mempengaruhi yaitu: ruangan yang tersedia, jumlah mata kuliah yang diselenggarakan, waktu yang ada dan ketersediaan dosen yang mengajar. Oleh karena itu tujuan dari penelitian ini adalah merancang suatu sistem yang dapat membuat atau menyusun jadwal perkuliahan secara teroptimasi. Metode dalam proses pembuatan jadwal perkuliahan secara otomatis pada penelitian ini menggunakan metode algoritma genetika dengan teknik seleksi turnamen. Hasil pengujian sistem dapat memberikan kemudahan dan kecepatan kepada user atau Program Studi Teknologi Informasi dalam proses pembuatan atau penyusunan jadwal untuk perkuliahan, yaitu hanya diperlukan waktu sekitar 14,7 menit dibandingkan dengan proses manual yang memerlukan waktu sekitar 2 (dua) hari.

Kata kunci: Algoritma genetika, penjadwalan perkuliahan di Universitas, teknik seleksi turnamen

Abstract

For a college, the university course timetabling is an activity that's very important for the implementation of good teaching and learning process. In teaching and learning process can be done by all related parties, not only for Lecturers who teach, but also for students who take the course. In the preparation of the schedule, there are several variables that affect the: the available space, the number of courses held, the time available and the availability of lecturers who teach. Therefore, the purpose of this research is to design a system that can create or arrange optimization schedule optimally. Methods in the process of making university course timetabling automatically in this study using genetic algorithm method with tournament selection.

Keywords: Genetic Algorithm, University Course Timetabling, Tournament Selection

1. PENDAHULUAN

Penjadwalan perkuliahan merupakan suatu kegiatan yang sangat penting untuk dapat terlaksananya proses belajarmengajar yang baik. Dimana dalam proses belajar mengajar dapat dilakukan oleh semua pihak yang terkait, bukan hanya bagi dosen yang mengajar, tetapi juga bagi mahasiswa yang mengambil mata kuliah. Dalam proses penyusunan jadwal perkuliahan terdapat beberapa hal yang rumit dan sering mengalami kesulitan dalam penyusunannya. Seperti pengalokasian mata kuliah, dosen dan ruangan yang akan digunakan untuk perkuliahan agar tidak terjadi bentrok dengan jadwal mata kuliah, dosen dan ruang yang lain dalam satu periode jadwal kuliah (Science and Engineering, 2017; Er, Sayed and Ahmed, 2015).

Proses pembuatan jadwal perkuliahan di Program Studi Teknologi Informasi Fakultas Teknik

Universitas Lambung Mangkurat yang ada saat ini dikelola oleh admin dengan persetujuan semua dosen yang mengajar dan Kepala Program Studi Teknologi Informasi. Karena saat ini proses pembuatan jadwal masih secara manual dengan menggunakan Microsoft Office Excel, Sehingga mengakibatkan waktu pekerjaan yang cukup lama dengan kisaran waktu sekitar 1 - 2 hari dari proses rapat sampai proses pembuatan jadwal selesai dan terdapat kesulitan dalam pengalokasian mata kuliah, dosen, mahasiswa yang mengambil mata kuliah beserta ruangnya agar tidak terjadi bentrok.

Optimasi dalam penyusunan jadwal perkuliahan adalah dimana dalam penyusunan jadwal perkuliahan terdapat suatu cara atau sistem yang mampu memberikan kecepatan, kemudahan dan hasil yang sesuai dengan batasan-batasan yang telah ditentukan dalam penyusunan jadwal perkuliahan (Salvi, Khanvilkar and Balkhande, 2016; Sahoo et al., 2017). Seperti ruangan tidak dapat digunakan dalam waktu

yang bersamaan. dosen tidak dapat mengajar dalam waktu yang bersamaan. tidak ada waktu perkuliahan yang lewat dari jam 18.00, tidak ada proses belajar mengajar pada waktu 12.10 sampai 13.00 karena waktu tersebut dialokasikan untuk ISHOMA, jumlah mahasiswa yang mengambil mata kuliah tidak boleh lebih dari kapasitas ruangan yang digunakan dan tidak ada proses belajar mengajar untuk mata kuliah yang semester dan kelasnya sama terjadi dalam waktu yang bersamaan.

Penelitian yang dibuat oleh Salwani Abdullah dan Hamza Turabieh dengan judul “*Generating University Course Timetable Using Genetic Algorithms and Local Search*” menyebutkan bahwa masalah penjadwalan perkuliahan di universitas adalah masalah yang sangat susah diselesaikan dengan menggunakan metode tradisional. Sejumlah besar metode yang beragam telah diajukan dalam literatur untuk memecahkan masalah pengaturan waktu. Metode ini berasal dari sejumlah disiplin ilmu seperti *Operations Research*, *Artificial Intelligence*, dan *Computational Intelligence* dan dibagi kedalam 4 kategori yaitu *Sequential Methods*, *Constraint Based Methods*, *Cluster Methods* dan *Meta-heuristic methods*. Algoritma genetika termasuk dalam kategori *Meta-heuristic methods*. Algoritma Genetika telah digunakan dalam pemecahan masalah penjadwalan sejak 1990 dan banyak penelitian menyajikan metode dan aplikasi evolusioner mengenai masalah penjadwalan dengan keberhasilan yang signifikan. Penelitian ini membandingkan dengan pendekatan dari penelitian lain seperti *Iterative Improvement Algorithm*, *a graph hyperheuristic* dan *variable neighbourhood search* (VNS). hasil dari algoritma genetika menghasilkan jadwal yang dapat diterima & Turabieh, 2008).

Sebuah solusi yang optimal didefinisikan sebagai pemanfaatan terbaik dari sebuah keahlian, sumber daya dan waktu yang tersedia untuk mendapatkan hasil yang terbaik. Pemanfaatan terbaik didefinisikan di bawah serangkaian *constraints*. *Constraints* dibagi menjadi hard dan soft constraintss. Dalam penelitian berjudul “*Developing A Course Timetable System For Academic Departments Using Genetic Algorithm*” yang dibuat oleh Mohammad A. Al-Jarrah, Ahmad A. Al-Sawalqah dan Sami F. Al-Hamdan, *soft* dan *hard constraintss* yang digunakan yang berkaitan dengan waktu, ruang, dosen dan perkuliahan. Semua *constraints* telah dipertimbangkan dalam algoritma yang diusulkan untuk mengembangkan jadwal perkuliahan yang sesuai dengan sedikit usaha. Dalam Penelitian ini metode yang digunakan adalah algoritma genetika, karena algoritma genetika merupakan sebuah kelas teknik non-tradisional yang dapat menangani ruang pencarian yang kompleks dan besar sampai batas tertentu untuk pemecahan suatu masalah (Al-Jarrah, dkk, 2017; Shukla, dkk, 2015). Dalam beberapa untuk mengatasi permasalahan penyusunan jadwal perkuliahan dan berdasarkan paparan diatas maka

penulis mengusulkan penelitian yang berjudul “Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik Tournament Selection “. Tempat Penelitian yang digunakan adalah di Program Studi Teknologi Informasi Fakultas Teknik Universitas Lambung Mangkurat. Metode algoritma genetika digunakan untuk menghasilkan jadwal perkuliahan karena algoritma genetika merupakan salah satu metode yang dapat memecahkan permasalahan penyusunan jadwal perkuliahan.

2. LANDASAN TEORI

A. Penjadwalan

Penjadwalan adalah kegiatan pengalokasian yang konsisten antara ruangan, dosen dan mahasiswa dengan pengaturan yang terbaik (Rudová, 2014).

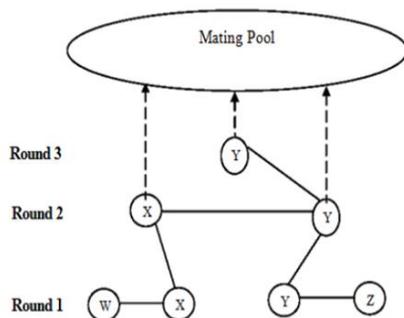
Kendala utama dalam penyusunan jadwal yang terjadi hampir diseluruh perguruan tinggi adalah jadwal bentrok. Misalnya, dosen tidak dapat berada di dua tempat sekaligus. Ini termasuk kendala berat menurut Lukas dkk. Kendala ringan adalah kendala yang mungkin rusak, tetapi dari pelanggaran yang harus diminimalkan. Sebagai contoh, ada pesanan jadwal dari dosen bersangkutan (Lukas dkk, 2012). Selain kendala tersebut ada sejumlah syarat dalam membuat sistem penjadwalan otomatis. Dalam penelitian ini, kendala yang sulit adalah ruang kelas dimana kelas untuk praktikum hanya ada dua laboratorium dan ruang laboratorium hanya bisa di gunakan oleh 30 orang peserta kelas. Sedangkan kendala yang ringan adalah beberapa dosen memiliki permintaan jam yang dipilih untuk dijadwalkan.

B. Algoritma Genetika

Algoritma genetika adalah algoritma pencarian dan teknik optimasi yang memiliki sejumlah langkah seperti inisialisasi, seleksi, *crossover*, mutasi dan penggantian. Algoritma genetika adalah algoritma optimasi pencarian yang memaksimalkan atau meminimalkan fungsi yang diberikan. Awalnya algoritma genetika memiliki prosedur heuristik, algoritma genetika tidak dijamin untuk menemukan yang optimal, tetapi pengalaman telah menunjukkan bahwa algoritma genetika dapat menemukan solusi yang sangat baik untuk berbagai masalah (Yadav & Sohal, 2017). Di penelitian ini algoritma genetika menggunakan teknik *Tournament Selection*.

Algoritma genetika menggunakan mekanisme pemilihan untuk memilih individu dari populasi untuk dimasukkan ke dalam kolam kawin. Individu dari kolam kawin digunakan untuk menghasilkan keturunan baru, dengan pembentukan keturunan yang dihasilkan sebagai dasar generasi berikutnya. Mekanisme pemilihan dalam proses pembentukan keturunan adalah algoritma genetika akan memilih individu dalam populasi yang lebih baik terlebih dahulu dari kolam kawin. Tekanan seleksi adalah sejauh mana individu yang lebih baik disukai:

semakin tinggi tekanan ion pilih, semakin banyak individu yang lebih baik disukai. Tekanan seleksi ini mendorong algoritma genetika untuk meningkatkan *fitness* populasi atas generasi berikutnya. Tingkat konvergensi algoritma genetika sangat ditentukan oleh tekanan seleksi, dengan tekanan seleksi yang lebih tinggi menghasilkan tingkat konvergensi yang lebih tinggi. Namun, jika pemilihan tekanan terlalu rendah, tingkat konvergensi akan lambat, dan algoritma genetika tidak perlu lebih lama untuk menemukan solusi optimal. Jika tekanan seleksi terlalu tinggi, ada kemungkinan peningkatan algoritma genetika sebelumnya menyatu menjadi salah satu solusi optimal (suboptimal). *Tournament Selection* memberikan tekanan dengan memegang sebuah turnamen di antara para pesaing, dengan menjadi ukuran turnamen. Pemenang turnamen adalah individu dengan *fitness* tertinggi dari kompetitor turnamen. Pemenangnya kemudian dimasukkan ke dalam kolam kawin. Pada kolam kawin, yang terdiri dari pemenang turnamen, memiliki *fitness* rata-rata yang lebih tinggi daripada *fitness* populasi rata-rata. Perbedaan *fitness* ini memberikan tekanan pemilihan, yang mendorong algoritma genetika untuk meningkatkan kebugaran setiap generasi berikutnya. Meningkatkan tekanan pilihan dapat diberikan dengan mudah meningkatkan ukuran s turnamen, sebagai pemenang dari yang lebih besar turnamen, rata-rata, memiliki *fitness* yang lebih tinggi dari pada pemenang turnamen yang lebih kecil (Yadav& Sohal, 2017; Lukas, Aribowo and Muchri, 2012). Ilustrasi kolam kawin dapat dilihat pada gambar 1.



Gambar 1. Ilustrasi Kolam Kawin

a. Komponen-komponen Algoritma Genetika

Menurut Puspaningrum, ada 6 komponen utama yang terdapat di dalam algoritma genetika (Puspaningrum, dkk, 2013), yaitu:

1. Teknik Pengkodean

Teknik pengkodean merupakan suatu teknik bagaimana mengkodekan gen dari kromosom. Teknik ini merupakan teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok masalah. Teknik pengkodean meliputi pengkodean gen dan kromosom. Gen adalah bagian dari kromosom yang dapat direpresentasikan dalam bentuk string bit, tree,

array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lain yang dapat diimplementasikan untuk operator genetika.

2. Prosedur Inisialisasi (*generate* populasi awal)

Proses ini menghasilkan sejumlah individu secara acak (*random*). Banyaknya populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diterapkan. Setelah jumlah populasi ditentukan, selanjutnya dilakukan inisialisasi terhadap kromosom yang ada di dalam populasi tersebut. Inisialisasi kromosom dilakukan secara acak, dengan tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

3. Fungsi Evaluasi

Individu dievaluasi berdasarkan fungsi tertentu sebagai ukuran kinerjanya. Individu dengan nilai *fitness* tinggi pada kromosomnya yang akan dipertahankan, sedangkan individu yang pada kromosomnya bernilai *fitness* rendah akan diganti. Fungsi *fitness* tergantung pada permasalahan tertentu dari representasi yang digunakan. Perhitungan nilai *fitness* dari setiap kromosom dapat dirumuskan sebagai berikut:

$$\text{Fitness} = \frac{1}{1+(F1B1+F2B2+\dots)}$$

Keterangan :

Bn = Bobot pelanggaran

Fn = Banyaknya pelanggaran

n = 1 . . . n

4. Seleksi

Seleksi yang digunakan dalam penyelesaian algoritma genetika ini yaitu dengan menggunakan teknik *Tournament Selection*. *Tournament Selection* digunakan karena efisiensi dan kemudahan implementasi, *Tournament Selection* adalah teknik seleksi genetik yang paling populer. Dalam *Tournament Selection*, individu dari keseluruhan populasi bersaing satu sama lain. Individu dengan nilai *fitness* tertinggi menang dan dipilih untuk diproses lebih lanjut.

5. Operator Genetika

Algoritma genetika merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Ada dua operator genetika yaitu:

a. *Crossover* (Persilangan)

Crossover merupakan proses di dalam algoritma genetika yang bekerja untuk menggabungkan dua kromosom *parent* menjadi sebuah kromosom baru (*offspring*) pada suatu waktu. Sebuah kromosom yang akan mengarah pada solusi baik dapat diperoleh melalui proses *crossover* dari hasil persilangan dua buah kromosom. Cara sederhana pada proses *crossover* yaitu dengan memilih satu titik (*one cut-point*) yang dipisahkan secara acak dan kemudian

membentuk *offspring* dengan mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan.

b. *Mutation* (Mutasi)

Merupakan proses untuk merubah salah satu atau lebih beberapa gen dari suatu kromosom. Proses ini berfungsi untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inialisasi populasi.

6. Parameter Kontrol

Parameter kontrol diperlukan untuk mengendalikan operator-operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah. Ada dua parameter dasar dari algoritma genetika, yaitu probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm).

a. Probabilitas *Crossover* (Pc)

Probabilitas *crossover* akan mengendalikan operator *crossover* dalam setiap generasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Apabila nilai probabilitas *crossover* terlalu besar, maka individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya.

b. Probabilitas mutasi (Pm)

Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasi dengan peluang mutasi yang digunakan lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul, sehingga operator mutasi pada algoritma genetik tidak selalu terjadi (Suhartono, 2015).

3. METODE PENELITIAN

Data yang digunakan dalam penelitian ini berupa data sekunder, yaitu berupa jadwal perkuliahan Program Studi Teknologi Informasi Fakultas Teknik Universitas Lambung Mangkurat semester ganjil tahun ajaran 2017/2018,

Pada penelitian ini pengumpulan data dilakukan dengan metode studi literatur. Data didapat dan dikumpulkan dari jurnal, skripsi, buku yang berhubungan dengan topik penelitian ini dan jadwal yang sudah ada di Program Studi Teknologi Informasi. Data untuk penelitian ini menggunakan data sekunder yang didapat dari beberapa jurnal tentang pembuatan jadwal perkuliahan secara otomatis dan jadwal perkuliahan di Program Studi Teknologi Informasi.

Data yang digunakan untuk proses generate dengan menggunakan algoritma genetika adalah data perkuliahan Program Studi Teknologi Informasi Universitas Lambung Mangkurat semester ganjil tahun ajaran 2017/2018.

Pengumpulan data yang telah dilakukan menghasilkan data yang diperlukan sebagai masukan, yaitu:

Tabel 1. Rekapitulasi Data

No	Keterangan	Jumlah Data
1	Jumlah ruangan yang digunakan	7 Ruangan
2	Jumlah daftar mata kuliah semester ganjil tahun ajaran 2017/2018 di Program Studi Teknologi Informasi	26 Daftar Mata Kuliah
3	Jumlah keseluruhan dosen	13 Dosen
4	Jumlah waktu	60 waktu

Dalam penyusunan penjadwalan terdapat batasan kaku (*hard constraintss*) dan batasan lunak (*soft constraintss*) (Al-Jarrah, dkk, 2017; Puspaningrum, dkk, 2013; Suhartono, 2015) seperti berikut ini:

a. Batasan kaku (*hard constraintss*):

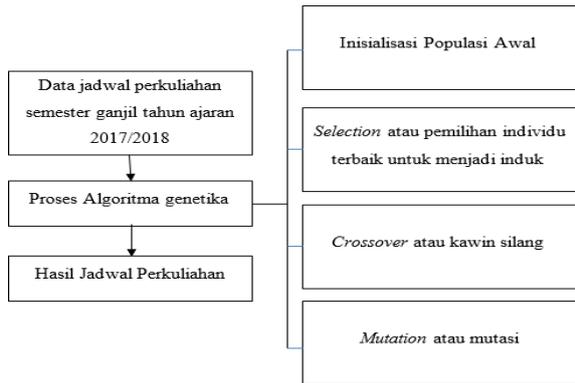
- Ruang tidak dapat digunakan dalam waktu yang bersamaan.
- Dosen tidak dapat mengajar dalam waktu yang bersamaan.
- Tidak ada waktu perkuliahan yang lewat dari jam 18.00
- Tidak ada proses belajar mengajar pada waktu 12.10 - 13.00 karena waktu tersebut dialokasikan untuk ISHOMA.
- Jumlah mahasiswa yang mengambil mata kuliah tidak boleh lebih dari kapasitas ruangan yang digunakan.
- Mata kuliah dengan semester dan kelas yang sama tidak dapat dilaksanakan di waktu yang bersamaan.
- Mata kuliah yang bersifat non-praktikum tidak dapat menggunakan ruangan praktikum.
- Mata kuliah yang bersifat praktikum tidak dapat menggunakan ruangan non-praktikum.

b. Batasan lunak (*soft constraintss*):

- Dosen dapat melakukan request waktu untuk mengajar yang terdiri dari pagi (08:00, 08:50, 09:40, 10:30, 11:20) dan siang (13:00, 13:50, 14:40, 15:30, 16:20, 17:10) setiap harinya.
- Dosen dapat melakukan request waktu dengan catatan harus memilih waktu minimal sebanyak jumlah mata kuliah yang diampu atau diajarkan.

Adapun rancangan alur sistem yang akan dibuat dapat dilihat pada Gambar 2 berikut.

Dari Gambar 2 dapat dijelaskan proses dari pembuatan jadwal perkuliahan. Tahap pertama yaitu input data jadwal perkuliahan meliputi data ruangan, data dosen, data mata kuliah dan jumlah mahasiswa yang mengambil mata kuliah. Setelah diinput, data kemudian di proses dengan metode algoritma genetika. Proses selanjutnya diketahui hasil jadwal perkuliahan (Duan et al., 2018; Shinde and Gurav, 2018; Kooy, 2017). Jika kegiatan sudah mendapatkan hasil yang sesuai atau sudah mencapai nilai maksimal generasi maka proses selesai.



Gambar 2 Rancangan Alur Sistem

4. HASIL DAN PEMBAHASAN

Proses pembuatan jadwal perkuliahan secara otomatis menggunakan metode algoritma genetika dengan tournament selection dilakukan dengan menggunakan 7 data ruangan, 26 daftar mata kuliah, 13 dosen dan 60 waktu. Proses pengujian model dilakukan dengan spesifikasi perangkat keras dan perangkat lunak sebagai berikut:

- Intel Core i7 Processor
- RAM 8 GB
- Microsoft Windows 10 64-bit
- Laravel 5.5
- PHP 7.2.1

Untuk mengetahui waktu dan akurasi terbaik yang bisa diraih maka harus dilakukan uji coba sistem terlebih dahulu dengan melakukan 5x (lima kali) percobaan untuk setiap nilai variabel yang diinputkan sehingga diketahui rata-rata estimasi waktu dan nilai fitnessnya. Untuk percobaan pertama yang dilakukan adalah mencari nilai variabel terbaik yang sesuai dengan hard constraintss dengan langkah pertama mencari nilai variabel maksimal generasi (Sani, 2016). Adapun hasil pengujian dari nilai variabel maksimal generasi menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada Tabel 2 berikut ini:

Tabel 2 Hasil Pengujian Dari Nilai Variabel Maksimal Generasi Berdasarkan Hard constraintss

No	Jml Pop Dibangkitkan	Maks Gene rasi	Prob. Crossover (pc)	Prob. mutasi (pm)	Estimasi Wkt	Nilai Fitness
1	25	50	75	10	104,493 detik	0,497
2	25	100	75	10	115,453 detik	1

Berdasarkan Tabel 2 maka diketahui bahwa nilai variabel maksimal generasi agar mendapatkan nilai fitness dan estimasi waktu terbaik adalah 100, dimana estimasi waktu terbaik yang bisa diraih dengan catatan nilai fitness 1 (satu) adalah 115,453 detik atau sekitar 1,9 menit. Setelah mengetahui nilai variabel maksimal generasi yang terbaik, yang harus dilakukan adalah mencari nilai jumlah populasi yang

dibangkitkan. Adapun hasil pengujian dari nilai variabel jumlah populasi dibangkitkan menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada Tabel 3 berikut ini:

Tabel 3 Hasil Pengujian Dari Nilai Variabel Jumlah Populasi Dibangkitkan Berdasarkan Hard constraints

No	Jumlah Populasi Dibangkitkan	Maks Gene rasi	prob crossover (pc)	Prob mutasi (pm)	Estimasi Waktu	Nilai Fitness
1	20	100	75	10	162.801 detik	0,554
2	25	100	75	10	115.453 detik	1
3	30	100	75	10	136.118 detik	1

Berdasarkan Tabel 3 maka diketahui bahwa nilai variabel jumlah populasi dibangkitkan agar mendapatkan nilai fitness dan estimasi waktu terbaik adalah 25, dimana estimasi waktu terbaik yang bisa diraih dengan catatan nilai fitness 1 (satu) adalah 115.453 detik atau sekitar 1,9 menit. Setelah mengetahui nilai variabel jumlah populasi dibangkitkan yang terbaik, yang harus dilakukan adalah mencari nilai probabilitas crossover (pc). Adapun hasil pengujian dari nilai variabel probabilitas crossover (pc) menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada Tabel 4 berikut ini:

Tabel 4 Hasil Pengujian Dari Nilai Variabel Probabilitas Crossover (PC) Berdasarkan Hard constraints

No	Jml Pop. Dibangkitkan	Maks Generasi	Prob. crossover (pc)	Prob mutasi (pm)	Estimasi Waktu	Nilai Fitness
1	25	100	70	10	211.735 detik	0,333
2	25	100	75	10	115.453 detik	1
3	25	100	80	10	123.049 detik	1

Berdasarkan Tabel 4 maka diketahui bahwa nilai variabel probabilitas crossover (pc) agar mendapatkan nilai fitness dan estimasi waktu terbaik adalah 75, dimana estimasi waktu terbaik yang bisa diraih dengan catatan nilai fitness

1 (satu) adalah 115.453 detik atau sekitar 1,9 menit. Setelah mengetahui nilai variabel probabilitas crossover (pc) yang terbaik, yang harus dilakukan adalah mencari nilai probabilitas mutasi (pm). Adapun hasil pengujian dari nilai variabel probabilitas mutasi (pm) menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada Tabel 5 berikut ini

Berdasarkan Tabel 5 maka diketahui bahwa nilai variabel probabilitas mutasi (pm) agar mendapatkan nilai fitness dan estimasi waktu terbaik adalah 5 dimana estimasi waktu terbaik yang bisa diraih dengan catatan nilai fitness 1 (satu) adalah 109.270 detik atau sekitar 1,8 menit.

Tabel 5 Hasil Pengujian Dari Nilai Variabel Probabilitas Mutasi (PM) Berdasarkan *Hard constraints*

No	Jumlah Pop. Dibangkitkan	Maks Generasi	Probab <i>Crossover</i>	Probab mutasi	Estimasi Waktu	Nilai Fitness
1	25	100	75	3	111.833 detik	1
2	25	100	75	5	109.270 detik	1
3	25	100	75	10	115.453 detik	1
4	25	100	75	15	279.055 detik	0,11 1

Setelah mengetahui nilai variabel jumlah populasi dibangkitkan, maksimal generasi, probabilitas *crossover* dan probabilitas mutasi yang terbaik, maka diketahui nilai-nilai variabel yang dapat digunakan agar menghasilkan nilai fitness dan estimasi waktu terbaik, Adapun nilai-nilai variabel terbaik berdasarkan *hard constraints* untuk jumlah populasi dibangkitkan adalah 25, maksimal generasi adalah 100, probabilitas *crossover* adalah 75 dan probabilitas mutasi adalah 5.

Berdasarkan hasil pengujian yang tertera pada Tabel 5 masih belum melakukan perhitungan untuk *soft constraints*, dimana dosen belum dapat melakukan *request* waktu mengajar dengan catatan minimal waktu yang dipilih harus sebanyak jumlah mata kuliah yang diampu. Adapun contoh skenario yang digunakan dapat dilihat pada Tabel 6 berikut ini:

Tabel 6 Skenario *Request* Waktu

No	Nama Dosen	<i>Request</i> Waktu
1	Muhammad Alkaff, S.Kom., M.Kom	Senin - Siang
2	Eka Setya Wijaya, S.Kom., M.Kom	Senin - Siang, Kamis - Siang, Jumat - Siang
3	Husnul Khatimi, MT	Selasa - Pagi, Selasa - Siang, Jumat - Siang
4	Juhriyansyah Dalle, M.Kom., Ph.D	Rabu - Pagi, Rabu - Siang, Kamis - Siang
5	Muhammad Alkaff, S.Kom., M.Kom	Senin - Pagi, Kamis - Pagi, Jumat - Pagi
6	Yuslena Sari, S.Kom., M.Kom	Senin - Pagi, Selasa - Pagi, Rabu - Pagi, Kamis - Pagi

Tahap awal dalam percobaan agar mengetahui nilai variabel yang sesuai untuk *hard constraints* dan *soft constraints* dimulai dengan hanya menggunakan 1 (satu) dosen yang melakukan *request* waktu mengajar, dimana dalam percobaan ini yang digunakan adalah data dosen Yuslena Sari, S.Kom., M.Kom yang mengampu 4 (empat) mata kuliah perminggunya dengan *request* waktu mengajar Senin-Pagi, Selasa-Pagi dan Rabu-Pagi. Adapun berdasarkan hasil ujicoba didapatkan bahwa nilai-nilai variabel yang digunakan agar mendapatkan kombinasi terbaik dapat dilihat pada Tabel 7 berikut ini:

Tabel 7 Hasil Pengujian Dari Nilai Variabel Berdasarkan *Hard constraints* dan *Soft constraints* (1 Dosen)

No	Jumlah Pop. Dibangkitkan	Maks Generasi	Probab <i>Crossover</i> (pc)	Probab mutasi (pm)	Estimasi Waktu	Nilai Fitness
1	25	100	75	5	139.415 detik	0,48 3
2	25	150	75	5	228.940 detik	0,56 7
3	25	200	75	5	252.88 detik	1

pada inputan pertama dengan menggunakan maksimal generasi 100 dengan percobaan sebanyak 5x (lima kali) didapatkan nilai rata-rata fitness yaitu 0,733 yang menandakan ada mata kuliah yang tidak sesuai dengan request waktu dosen mengajar, sedangkan inputan ke 2 menggunakan generasi 150 dengan percobaan sebanyak 5x (lima kali) didapatkan nilai rata-rata fitness yaitu 0,567 yang menandakan jadwal sudah sesuai berdasarkan request waktu dosen untuk mengajar dengan estimasi waktu 228.940 detik atau sekitar 3,8 menit, sedangkan inputan ke 3 menggunakan generasi 200 dengan percobaan sebanyak 5x (lima kali) didapatkan nilai rata-rata fitness yaitu 1 yang menandakan jadwal sudah sesuai berdasarkan request waktu dosen untuk mengajar dengan estimasi waktu 252.88 detik atau sekitar 4,2 menit. Adapun berdasarkan hasil ujicoba didapatkan bahwa nilai-nilai variabel yang digunakan agar mendapatkan kombinasi terbaik untuk 1 request dosen yang mengajar 4 mata kuliah untuk jumlah populasi dibangkitkan adalah 25, maksimal generasi adalah 200, probabilitas *crossover* adalah 75 dan probabilitas mutasi adalah 5.

Setelah melakukan dan mendapatkan hasil pengujian terbaik untuk *hard constraints* dan *soft constraints* dengan hanya menggunakan 1 (satu) dosen yang melakukan *request* waktu mengajar. Selanjutnya dilakukan ujicoba dengan menggunakan 3 (tiga) dosen yaitu Husnul Khatimi, MT, Muhammad Alkaff, S.Kom., M.Kom dan Yuslena Sari, S.Kom., M.Kom. Adapun berdasarkan hasil ujicoba didapatkan bahwa nilai-nilai variabel yang digunakan agar mendapatkan kombinasi terbaik dapat dilihat pada Tabel 8 berikut ini:

Tabel 8 Hasil Pengujian Dari Nilai Variabel Berdasarkan *Hard constraints* dan *Soft constraints* (3 Dosen)

No	Jml Pop. Dibangkitkan	Maks Generasi	Prob. <i>Cross Over</i> (pc)	Prob. Muta -si (pm)	Estimasi Waktu	Nilai Fitness
1	25	200	75	5	287.939 detik	0,333
2	25	300	75	5	457.939 detik	0,5
3	25	500	75	5	629.480 detik	1

Berdasarkan hasil pengujian yang tertera pada Tabel 8, didapatkan hasil estimasi waktu dan nilai fitness berdasarkan *hard constraints* dan 3 (tiga) dosen melakukan *request* waktu. Adapun berdasarkan hasil ujicoba didapatkan bahwa nilai-nilai variabel yang

digunakan agar mendapatkan kombinasi terbaik untuk jumlah populasi dibangkitkan adalah 25, maksimal generasi adalah 500, probabilitas *crossover* adalah 75 dan probabilitas mutasi adalah 5 dengan estimasi waktu sekitar 10,5 menit.

Setelah mendapatkan hasil pengujian terbaik untuk *hard constraints* dan *soft constraints* dengan hanya menggunakan 3 (tiga) dosen yang melakukan *request* waktu mengajar. Selanjutnya dilakukan uji coba dengan menggunakan 6 (enam) dosen yaitu Husnul Khatimi, MT, Muhammad Alkaff, S.Kom., M.Kom, Yuslena Sari, S.Kom., M.Kom, Dwi Hastuti, S.Kom., M.T, Eka Setya Wijaya, S.Kom., M.Kom, Juhriyansyah Dalle, M.Kom., Ph.D. Adapun berdasarkan hasil uji coba didapatkan bahwa nilai-nilai variabel yang digunakan agar mendapatkan kombinasi terbaik dapat dilihat pada Tabel 9 berikut ini:

Tabel 9 Hasil Pengujian Dari Nilai Variabel Berdasarkan *Hard constraints* dan *Soft constraints* (6 Dosen)

N o	Jumlah Populasi Dibangkitkan	Maksi mal Generasi	Probabi litas <i>cross over</i> (pc)	Probabi litas mutasi (pm)	Estimasi Waktu	Nilai Fitness
1	25	500	75	5	744.686 detik	0,333
2	25	550	75	5	586,233 detik	0,8
3	25	600	75	5	880.430 detik	1

Berdasarkan hasil pengujian yang tertera pada Tabel 9, didapatkan hasil estimasi waktu dan nilai fitness berdasarkan *hard constraints* dan 6 (enam) dosen melakukan *request* waktu. Adapun berdasarkan hasil uji coba didapatkan bahwa nilai-nilai variabel yang digunakan agar mendapatkan kombinasi terbaik untuk jumlah populasi dibangkitkan adalah 25, maksimal generasi adalah 600, probabilitas *crossover* adalah 75 dan probabilitas mutasi adalah 5, dimana estimasi waktu terbaik yang bisa diraih dengan catatan nilai fitness 1 (satu) adalah 880.430 detik atau sekitar 14,7 menit.

5. KESIMPULAN

Dari hasil pengujian sistem dapat memberikan kemudahan dan kecepatan kepada user atau Program Studi Teknologi Informasi (Admin) dalam proses pembuatan atau penyusunan jadwal untuk perkuliahan, yaitu hanya diperlukan waktu sekitar 14,7 menit dibandingkan dengan proses manual yang memerlukan waktu sekitar 2 (dua) hari. Sistem dapat menghasilkan jadwal perkuliahan, sehingga mendapatkan kombinasi terbaik antara mata kuliah, dosen yang mengajar, mahasiswa yang mengambil mata kuliah, dan ruangan yang tersedia. Hasil dari pengujian didapatkan nilai variabel jumlah populasi dibangkitkan adalah 25, maksimal generasi adalah 600, probabilitas *crossover* adalah 75 dan probabilitas mutasi adalah 5, adapun untuk *hard constraint* telah terpenuhi semua dengan estimasi waktu 109.270 detik atau sekitar 1,8 menit, sedangkan untuk *soft constraints* dengan 6 (enam) dosen melakukan *request* adalah 880,430 detik atau sekitar 14,7 menit.

Pengalokasian mata kuliah, dosen dan ruangan pada tidak ada bentrok jadwal jika nilai fitness sama dengan 1.

DAFTAR PUSTAKA

- ABDULLAH, S. AND TURABIEH, H., 2008. Generating University Course Timetable Using Genetic Algorithms and Local Search. pp.254–260.
- AL-JARRAH, M.A., AL-SAWALQAH, A.A. AND AL-HAMDAN3, S.F., 2017. Developing A Course Timetable System For Academic Departments Using Genetic Algorithm. 3(1), pp.25–36.
- DUAN, K., FONG, S., SIU, S.W.I., SONG, W. AND ID, S.S.G., 2018. SS symmetry Adaptive Incremental Genetic Algorithm for Task Scheduling in Cloud Environments. pp.1–13.
- ER, P., SAYED, S. AND AHMED, A., 2015. Automated Timetable Generator. 1(11), pp.118–121.
- KOOY, N.J. VAN DER, 2017. The High School Scheduling Problem: Improving Local Search.
- LUKAS, S., ARIBOWO, A. AND MUCHRI, M., 2012. Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: University of Pelita Harapan Timetable. Intech open, 2, p.16.
- PUSPANGRUM, W.A., DJUNAIDY, A. AND VINARTI, R.A., 2013. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. 2(1), pp.127–131.
- RUDOVÁ, H., 2014. University Course Timetabling: From Theory to Practice.
- SAHOO, R.K., OJHA, D., MOHAPATRA, D.P. AND PATRA, M.R., 2017. Automatic Generation And Optimization Of Course Timetable Using A Hybrid. 95(1).
- SALVI, A., KHANVILKAR, O. AND BALKHANDE, B.W., 2016. Automatic Time-Table Generation System using Genetic Algorithm. 5(3), pp.824–826.
- SANI, H.M., 2016. Solving Timetabling problems using Genetic Algorithm Technique. 134(15), pp.33–38.
- SCIENCE, C. AND ENGINEERING, S., 2017. Automatic Time Table Generator 1. 7(5), pp.204–211.
- SHINDE, S. AND GURAV, S., 2018. Automatic Timetable Generation using Genetic Algorithm. 9(4), pp.19–21.
- SHUKLA, A., PANDEY, H.M. AND MEHROTRA, D., 2015. Comparative Review of Selection Techniques in Genetic Algorithm. pp.515–519.
- SUHARTONO, E., 2015. Optimasi Penjadwalan Dengan Algoritma Genetika(Studi Kasus di AMIK JTC Semarang). pp.132–146.

YADAV, S.L. AND SOHAL, A., 2017. Comparative Study of Different Selection Techniques in Genetic Algorithm. *International Journal of Science and Mathematics Education*, 6(3), pp.251–254.