

PENYUSUNAN JADWAL ASISTEN PRAKTIKUM MENGGUNAKAN ALGORITMA GENETIKA

¹M. Dimas S. Sanapiah, ²Anim Rofi'ah, ³Heny Dwi Jayanti, ⁴Alysha Ghea Arliana,
⁵Vivi Nur Wijayaningrum

^{1,2,3,4,5}Fakultas Ilmu Komputer, Universitas Brawijaya

Email: dimassisop@gmail.com, animrof@gmail.com, 27heny@gmail.com, alysha.ghea@gmail.com,
vivinurw@gmail.com

(Diterima: 5 Mei 2019, direvisi: 14 Mei 2019, disetujui: 16 Mei 2019)

ABSTRAK

Penjadwalan asisten praktikum merupakan proses untuk merancang jadwal mengajar praktikum yang setiap semester dilakukan oleh beberapa asisten. Adanya aturan-aturan yang harus dipatuhi dalam penyusunan jadwal menyebabkan proses penjadwalan ini menjadi cukup sulit. Pada penelitian ini, algoritma genetika digunakan untuk melakukan optimasi penyusunan jadwal asisten praktikum. Data yang digunakan terdiri dari data jadwal praktikum di laboratorium dan jadwal kuliah asisten. Berdasarkan hasil pengujian parameter yang telah dilakukan, parameter optimal algoritma genetika terdiri dari ukuran populasi sebesar 10, banyaknya generasi sebesar 50, nilai crossover rate sebesar 0.7 dan nilai mutation rate sebesar 0.3. Dengan menggunakan parameter optimal tersebut, algoritma genetika mampu menyusun jadwal asisten praktikum tanpa melanggar aturan-aturan yang telah ditentukan, yaitu dengan menghasilkan solusi yang mempunyai nilai fitness sama dengan 1 (satu).

Kata Kunci: algoritma genetika, asisten praktikum, optimasi, penjadwalan

1 PENDAHULUAN

Penjadwalan merupakan sebuah proses mengalokasikan sumber daya untuk menentukan pekerjaan yang ditetapkan dalam suatu periode tertentu [1]. Seluruh sumber daya yang ada harus dialokasikan sesuai dengan waktu dan batasan yang telah ditentukan. Penjadwalan merupakan permasalahan kombinatorial yang cukup kompleks dan sulit diselesaikan menggunakan cara konvensional karena membutuhkan waktu yang lama untuk mendapatkan solusi yang optimal [2]. Penjadwalan harus diatur sistematis secara efektif dan efisien sehingga tujuan dapat tercapai secara maksimal.

Penjadwalan asisten praktikum merupakan salah satu contoh jenis penjadwalan yang melibatkan banyak parameter, misalnya ketersediaan laboratorium, jadwal asisten, jadwal praktikum mahasiswa (praktikan), dan sebagainya. Tujuan dari penjadwalan ini adalah menentukan jadwal setiap asisten praktikum untuk mengajar di beberapa kelas tertentu, dengan batasan berupa jumlah maksimal kelas yang boleh diajar, serta jadwal kuliah asisten dan jadwal praktikum tidak boleh bentrok. Beberapa batasan dalam penjadwalan harus diperhitungkan dengan baik untuk mempermudah proses penjadwalan yang baik. Apabila jadwal yang disusun tidak optimal, maka sumber daya manusia ataupun penggunaan laboratorium juga tidak akan optimal.

Pada permasalahan penjadwalan, terdapat dua jenis batasan atau kendala yaitu *hard constraint* dan *soft constraint*. *Hard constraint* adalah batasan yang tidak dapat dilanggar saat proses penyusunan jadwal [3]. Misalnya, jika seorang asisten sudah dijadwalkan untuk satu slot waktu tertentu, maka asisten tersebut harus tersedia di slot waktu tersebut. Sebuah solusi dapat diterima ketika tidak ada *hard constraint* yang dilanggar. Sedangkan *soft constraint* adalah batasan yang ingin ditangani sebanyak mungkin di dalam solusi. Misalnya asisten diberikan alokasi slot waktu untuk menentukan waktu secara bebas sesuai dengan yang diinginkan [4]. Dengan demikian, pada masalah penjadwalan, *hard constraint* harus dipatuhi dan dalam waktu yang bersamaan juga berusaha untuk memenuhi sebanyak mungkin *soft constraint*.

Beberapa metode telah banyak digunakan untuk menyelesaikan permasalahan penjadwalan. Permasalahan kombinatorial sering kali diselesaikan menggunakan algoritma *heuristic* dan *meta-heuristik*. Pada penelitian sebelumnya, *Particle Swarm Optimization* digunakan untuk penjadwalan

Sanapiah, Penyusunan Jadwal Asisten Praktikum Menggunakan Algoritma Genetika

masalah *job-shop* dengan mesin multi-guna. Setiap partikel solusi berisi urutan susunan operasi yang harus dilakukan menggunakan mesin berdasarkan prioritasnya. Solusi optimal dihitung dari banyaknya waktu yang diperlukan untuk menyelesaikan seluruh pekerjaan yang ada [5]. Kemudian penelitian yang lain menggunakan *Simulated Annealing* untuk melakukan optimasi pada penjadwalan panen tebu. Data yang digunakan meliputi jumlah hari untuk panen dan jarak kebun tebu yang digunakan untuk menentukan banyaknya truk untuk proses pengangkutan tebu ke pabrik. Solusi jadwal terbaik yang dipilih adalah jadwal yang mempunyai perbedaan jumlah tebu terkecil dan panen tebu tidak lebih dari 1900 ton [6].

Sebagai salah satu algoritma meta-heuristik, algoritma genetika merupakan algoritma yang cukup kuat dan mampu menyelesaikan berbagai permasalahan multi-objektif [7]. Beberapa penelitian yang menggunakan algoritma genetika adalah penjadwalan rute kapal [8], penyusunan jadwal kuliah [9], dan pendistribusian barang farmasi [10]. Algoritma genetika dapat bekerja dengan baik pada berbagai jenis optimasi, baik fungsi analitis maupun data numerik. Selain itu, algoritma genetika sederhana mampu memberikan solusi yang baik untuk beberapa permasalahan yang kompleks [11]. Oleh karena itu, penyusunan jadwal asisten praktikum pada penelitian ini akan diselesaikan menggunakan algoritma genetika. Solusi yang ditawarkan berupa susunan jadwal praktikum yang valid untuk asisten dengan tidak melanggar batasan-batasan yang telah ditentukan.

Algoritma genetika merupakan sebuah algoritma meta-heuristik yang digunakan untuk menyelesaikan permasalahan dengan cara mencari sebuah solusi terbaik dari beberapa solusi yang ditawarkan [12]. Cara kerja algoritma genetika terinspirasi dari evolusi biologi, dimana terdapat sebuah populasi yang berisi beberapa kromosom. Setiap kromosom tersebut mewakili sebuah solusi dari permasalahan yang akan diselesaikan. Kromosom-kromosom tersebut akan dievaluasi kualitasnya menggunakan nilai *fitness*. Nilai *fitness* setiap kromosom bisa jadi berbeda, tergantung dari susunan kromosomnya. Dengan adanya nilai *fitness* ini, kromosom-kromosom yang ada di dalam sebuah populasi dapat diseleksi agar kromosom terbaik dapat dipertahankan untuk proses di generasi berikutnya [13].

2 METODE PENELITIAN

Penelitian ini menggunakan data yang terdiri dari jadwal praktikum di laboratorium dan jadwal kuliah asisten. Asisten praktikum sebanyak 6 orang akan ditugaskan untuk mengajar pada 14 kelas dari hari Senin sampai dengan Jumat. Jadwal praktikum di laboratorium tidak boleh bentrok dengan jadwal kuliah dari asisten tersebut. Selain itu, setiap asisten hanya boleh mengajar sekali dalam sehari dan jumlah maksimal kelas yang diajar adalah 3 kelas. Data jadwal praktikum di laboratorium dan jadwal kuliah asisten ditunjukkan pada Tabel 1 dan Tabel 2.

Tabel 1. Jadwal Praktikum di Laboratorium

Hari	Jam	Kode
Senin	07.00	1
	09.30	2
	15.20	3
Selasa	07.00	4
	12.50	5
Rabu	07.00	6
	12.50	7
Kamis	07.00	8
	09.30	9
	12.50	10
Jumat	07.00	11
	09.30	12
	12.50	13
	15.20	14

Tabel 2. Jadwal Kuliah Asisten

Nama Asisten	Hari	Jam
A	Senin	09.30
	Rabu	15.20
	Kamis	09.30
	Kamis	12.50
	Jumat	07.00
	Jumat	14.30
B	Jumat	16.10
	Selasa	09.30
	Selasa	15.20
	Rabu	12.50
	Kamis	09.30
C	Kamis	12.50
	Rabu	07.00
	Rabu	12.50
	Kamis	07.00
	Kamis	09.30
D	Kamis	15.20
	Selasa	07.00
	Selasa	09.30
	Selasa	12.50
	Rabu	12.50
E	Kamis	07.00
	Selasa	09.30
	Rabu	12.50
	Kamis	09.30
F	Jumat	07.00
	Rabu	07.00
	Rabu	12.50
	Kamis	07.00
	Kamis	09.30
	Kamis	15.20

2.1 Representasi Kromosom

Representasi kromosom yang digunakan untuk merepresentasikan solusi penyusunan jadwal asisten praktikum berupa representasi *integer* yang berisikan huruf A sampai dengan F sesuai dengan jumlah asisten yaitu sebanyak 6. Setiap kromosom terdiri dari 14 gen yang mewakili nama asisten secara acak. Karena representasi kromosom yang digunakan bertipe *integer*, maka sebuah huruf dapat muncul beberapa kali pada satu kromosom. Gambar 1 menunjukkan representasi kromosom yang digunakan pada penelitian ini.

Gen	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Kromosom	A	C	B	D	E	A	B	F	E	E	D	F	A	B

Gambar 1. Representasi kromosom menggunakan *integer*

Berdasarkan contoh representasi kromosom pada Gambar 1, dapat diketahui bahwa asisten A mengajar 3 kelas yaitu jadwal ke-1, ke-6, dan ke-13, asisten B mengajar 3 kelas yaitu jadwal ke-3, ke-7, dan ke-14, asisten C mengajar 1 kelas yaitu jadwal ke-2, asisten D mengajar 2 kelas yaitu jadwal ke-4 dan ke-11, asisten E mengajar 3 kelas yaitu jadwal ke-5, ke-9, dan ke-10, dan asisten F mengajar 2 kelas yaitu jadwal ke-8 dan ke-12.

2.2 Crossover

Metode *crossover* yang digunakan adalah *one-cut poin crossover*. Metode *one-cut point crossover* merupakan metode *crossover* yang paling mudah dan sederhana [14]. Dua buah kromosom dipilih secara acak dari sebuah populasi untuk digunakan sebagai *parent*, kemudian satu titik potong ditentukan secara acak untuk digunakan sebagai batas pertukaran gen pada kedua kromosom tersebut. Pada penelitian ini, satu kali proses *crossover* menghasilkan dua buah *offspring*. Contoh proses *crossover* ditunjukkan pada Gambar 2.

<i>Parent 1</i>	A	B	C	D	E	F	A	C	B	D	E	F	A	B
<i>Parent 2</i>	F	E	D	C	B	A	F	A	C	E	D	A	F	A
<i>Offspring 1</i>	A	B	C	D	E	F	A	A	C	E	D	A	F	A
<i>Offspring 2</i>	F	E	D	C	B	A	F	C	B	D	E	F	A	B

Gambar 2. Crossover menggunakan one-cut point

Pada Gambar 2, terdapat satu titik potong yang digunakan untuk proses *crossover* yaitu pada titik ke-7. Untuk menghasilkan *offspring 1*, gen-gen pada *Parent 1* di sebelah kiri titik potong (gen ke-1 sampai dengan ke-7) disalin untuk diwariskan ke *offspring 1*, sedangkan gen-gen sisanya didapatkan dari gen-gen pada *Parent 2* di sebelah kanan titik potong (gen ke-8 sampai dengan ke-14). Cara yang sama juga dilakukan untuk pembentukan *offspring 2*, tetapi dengan mendahulukan *Parent 2* untuk disalin gen-gennya dan diikuti dengan *Parent 1*.

2.3 Mutasi

Mutasi merupakan salah satu proses reproduksi yang berfungsi untuk mengembalikan informasi genetik yang hilang untuk membantu proses eksplorasi di ruang pencarian [15]. Metode mutasi yang digunakan adalah *random mutation*. Cara kerja *random mutation* adalah dengan memilih sebuah kromosom secara acak untuk digunakan sebagai *parent*, kemudian menentukan satu gen yang akan dimutasi secara acak. Selanjutnya gen tersebut diacak ulang untuk mendapatkan nilai baru. Kromosom dengan nilai gen baru inilah yang akan dijadikan sebagai *offspring*. Contoh proses mutasi ditunjukkan pada Gambar 3.

<i>Parent</i>	A	B	C	A	B	C	F	D	D	B	E	B	A	C
<i>Offspring</i>	A	B	C	E	B	C	F	D	D	D	E	B	A	C

Gambar 3. Mutasi menggunakan random mutation

2.4 Fungsi Fitness

Kromosom-kromosom yang telah terbentuk diukur kualitasnya menggunakan fungsi *fitness* dengan melakukan pengecekan terhadap aturan/batasan yang dilanggar dalam pembentukan solusi. Nilai *fitness* dihitung dengan cara memberikan penalti untuk setiap pelanggaran aturan dalam penyusunan jadwal. Semakin banyak aturan yang dilanggar, maka bobot penalti yang diberikan akan semakin besar. Bobot penalti untuk setiap pelanggaran aturan ditunjukkan pada Tabel 3.

Tabel 3. Bobot Penalti Pelanggaran Aturan

Aturan	Bobot Penalti
Jadwal praktikum tidak bentrok dengan jadwal kuliah asisten	0.50
Setiap asisten hanya boleh mengajar sekali dalam sehari	0.30
Setiap asisten hanya boleh mengajar maksimal 3 kelas	0.25

Selanjutnya untuk menghitung nilai *fitness* digunakan Persamaan 1, dengan n menyatakan banyaknya pelanggaran yang dilakukan dan w menyatakan bobot penalti sesuai dengan Tabel 3.

$$Fitness = \frac{1}{1 + \sum n \times w} \quad (1)$$

2.5 Seleksi

Metode seleksi yang digunakan adalah *elitism selection*. *Elitism selection* merupakan metode seleksi yang bekerja dengan cara mencari dan mengurutkan kromosom-kromosom di dalam populasi dari nilai *fitness* tertinggi sampai terendah untuk dijadikan sebagai populasi baru pada generasi berikutnya. Kromosom yang mempunyai kualitas solusi yang baik, yaitu nilai *fitness*-nya tinggi maka mempunyai peluang yang lebih besar untuk lolos ke generasi selanjutnya [16].

3 HASIL DAN PEMBAHASAN

Untuk mengetahui nilai-nilai parameter optimal pada algoritma genetika, tiga jenis pengujian dilakukan pada penelitian ini yaitu pengujian ukuran populasi, pengujian banyaknya generasi, dan pengujian kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*). Pada setiap jenis pengujian terdapat beberapa skenario pengujian yang dilakukan dengan nilai parameter yang berbeda-beda. Setiap skenario pengujian diujikan sebanyak 10 kali dan dihitung nilai rata-ratanya. Nilai-nilai parameter optimal yang dihasilkan dari ketiga jenis pengujian yang dilakukan ditunjukkan pada Tabel 4.

Parameter	Nilai Optimal
Ukuran populasi	10
Banyaknya generasi	50
<i>Crossover rate</i> (<i>cr</i>)	0.7
<i>Mutation rate</i> (<i>mr</i>)	0.3

Dengan menggunakan parameter optimal algoritma genetika yang telah didapatkan dari hasil pengujian, sistem memberikan solusi yang baik dengan nilai *fitness* yang tinggi yaitu rata-rata sama dengan 1. Tabel 5 menunjukkan beberapa solusi yang diberikan oleh sistem ketika dijalankan dengan menggunakan parameter optimal.

Percobaan	Solusi	Nilai <i>Fitness</i>
1	B E C B A A F A F E F C D B	1
2	C E B B E D A E D C D F A B	1
3	B C D A C E F F D E F B E A	1
4	B D A E A A F E D F C D B E	1
5	A D E B F A F B D F C A D B	1

Pada Tabel 5 diketahui bahwa dengan menggunakan parameter optimal, solusi yang dihasilkan oleh sistem memberikan nilai *fitness* sama dengan 1 para lima kali percobaan. Berdasarkan Persamaan 1, nilai *fitness* sama dengan 1 menunjukkan bahwa tidak ada aturan yang dilanggar untuk proses penyusunan jadwal, atau dalam kata lain tidak ada perhitungan nilai penalti. Hal ini membuktikan bahwa sistem dapat memberikan solusi yang baik tanpa melanggar aturan-aturan yang sudah ditetapkan, yaitu tidak ada jadwal yang bentrok antara jadwal kuliah asisten dengan jadwal praktikum di laboratorium, setiap asisten mengajar dengan jumlah maksimal 3 kelas, dan setiap asisten mengajar hanya sekali dalam sehari.

4 KESIMPULAN

Algoritma genetika terbukti mampu menyelesaikan permasalahan penyusunan jadwal asisten praktikum di laboratorium dengan memberikan solusi yang baik. Hal ini dapat dilihat dari hasil percobaan yang dilakukan dengan menggunakan parameter optimal algoritma genetika, yaitu mampu menghasilkan solusi dengan nilai *fitness* sama dengan 1. Nilai *fitness* ini menyatakan bahwa tidak ada permasalahan atau pelanggaran aturan selama penyusunan jadwal sehingga kualitas solusi yang diberikan sangat baik. Representasi kromosom yang digunakan adalah representasi *integer* yang mewakili nama asisten yang mengajar di setiap jadwal praktikum. Parameter optimal yang didapatkan dari hasil pengujian yaitu ukuran populasi sebesar 10, banyaknya generasi sebesar 50, nilai *crossover rate* sebesar 0.7 dan nilai *mutation rate* sebesar 0.3.

Sanapiah, Penyusunan Jadwal Asisten Praktikum Menggunakan Algoritma Genetika

REFERENSI

- [1] K. R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*. John Wiley & Sons, Inc., 2009.
- [2] A. Rochman, “Penjadwalan Kuliah Menggunakan Metode Constraints Programming dan Simulated Annealing,” in *Seminar Nasional Aplikasi Teknologi Informasi*, 2012.
- [3] A. Nanda, M. P. Pai, and A. Gole, “An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach,” *Int. J. Mach. Learn. Comput.*, vol. 2, no. 4, pp. 492–495, 2012.
- [4] P. Hosein and S. Boodhoo, “Event Scheduling with Soft Constraints and On-Demand Re-Optimization,” in *IEEE International Conference on Knowledge Engineering and Applications*, 2016, pp. 62–66.
- [5] P. Pongchairerks and K. Voratas, “A Particle Swarm Optimization Algorithm on Job-Shop Scheduling Problems with Multi-Purpose Machines,” *Asia-Pacific J. Oper. Res.*, vol. 26, no. 2, pp. 161–184, 2009.
- [6] E. N. Afifah, Alamsyah, and E. Sugiharti, “Scheduling Optimization of Sugarcane Harvest Using Simulated Annealing Algorithm,” *Sci. J. Informatics*, vol. 5, no. 2, pp. 138–147, 2018.
- [7] A. M. Purnomo, D. Werdiastu, T. Raissa, R. Widodo, and V. N. Wijayaningrum, “Algoritma Genetika untuk Optimasi Komposisi Makanan Bagi Penderita Hipertensi,” *J. Teknol. dan Sist. Komput.*, vol. 7, no. 1, pp. 1–6, 2019.
- [8] V. N. Wijayaningrum and W. F. Mahmudy, “Optimization of Ship’s Route Scheduling Using Genetic Algorithm,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 2, no. 1, pp. 180–186, 2016.
- [9] D. Kristiadi and R. Hartanto, “Genetic Algorithm for Lecturing Schedule Optimization (Case Study: University of Boyolali),” *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 1, pp. 83–94, 2019.
- [10] F. Ramadhani, F. A. Fathurrachman, R. Fitriawanti, A. C. Rongre, and V. N. Wijayaningrum, “Optimasi Pendistribusian Barang Farmasi Menggunakan Algoritma Genetika,” *Kumpul. J. Ilmu Komput.*, vol. 5, no. 2, pp. 159–168, 2018.
- [11] V. N. Wijayaningrum and W. F. Mahmudy, “Fodder composition optimization using modified genetic algorithm,” *Indones. J. Electr. Eng. Informatics*, vol. 7, no. 1, pp. 67–74, 2019.
- [12] W. F. Mahmudy, “Optimization of Part Type Selection and Machine Loading Problems in Flexible Manufacturing System Using Variable Neighborhood Search,” *IAENG Int. J. Comput. Sci.*, vol. 42, no. 3, pp. 254–264, 2015.
- [13] T. Sutojo, E. Mulyanto, and V. Suhartanto, *Kecerdasan Buatan*. Yogyakarta: Andi, 2011.
- [14] S. Dündar and I. Şahin, “Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways,” *Transp. Res. Part C Emerg. Technol.*, vol. 27, pp. 1–15, 2013.
- [15] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. New York, USA: Springer, 2008.
- [16] F. Liu, S. Liang, and X. Xian, “Optimal Path Planning for Mobile Robot Using Tailored Genetic Algorithm,” *Telkomnika*, vol. 12, no. 1, pp. 1–9, 2014.