

Penerapan Algoritma Genetika Sistem Penjadwalan Mata Pelajaran pada SMA Bina Jaya Palembang

APPLICATION OF GENETIC ALGORITHMS SUBJECT SCHEDULING SYSTEM
AT SMA BINA JAYA PALEMBANG

Yunita Yunita^{*1}, Kanda Januar Miraswan², Dhea Andini³

^{1,2,3} Fakultas Ilmu Komputer, Universitas Sriwijaya,

e-mail: ^{*1}yunita@ilkom.unsri.ac.id, ²kandajm@ilkom.unsri.ac.id, ³dheaandini98@gmail.com

Abstrak

Sistem penjadwalan di SMA Bina Jaya Palembang masih dilakukan secara manual. Dalam membuat jadwal, Wakil Kepala Sekolah Bidang Kurikulum harus mempertimbangkan banyak aspek yang menghambat proses penjadwalan. Algoritma genetika dipilih dalam penelitian ini karena telah banyak digunakan dalam penyelesaian masalah berupa penjadwalan, algoritma ini menggunakan fitness sebagai tolak ukur apakah suatu individu dikategorikan sebagai individu yang baik atau tidak, pengujian sistem dilakukan dengan menguji beberapa parameter yang menghasilkan nilai fitness terbaik. Hasil uji coba menunjukkan bahwa nilai fitness mendekati 1.0 yang membuktikan sistem berhasil memenuhi kebutuhan yang ada.

Kata kunci — Algoritma Genetika, Penjadwalan, Optimasi

Abstract

SMA Bina Jaya Palembang's scheduling system is still manually made. When creating schedules, the Vice Principals of Curriculum have to consider many aspects which hinders the scheduling process. Genetic algorithm has been chosen in this study because it has been widely used in the solution of problems in the form of scheduling, this algorithm uses fitness as a benchmark for whether an individual is categorized as a good individual or not, system testing is done by testing several parameters which produce the best fitness value Test results show that the fitness value is close to 1.0, which proves the system was successful in meeting the existing needs.

Kata kunci — Genetic Algorithm, Scheduling, Optimization

1. PENDAHULUAN

Jadwal Pelajaran merupakan elemen penting dari Sistem Informasi sekolah. Di setiap sekolah, kegiatan rutin di awal tahun pelajaran adalah membuat atau menyusun suatu jadwal pelajaran. Dalam pembuatannya diperlukan keseriusan kerja, jika tidak maka jadwal yang dibuat tidak akan sempurna sehingga adanya bentrok antara guru dan juga ruang kelas dalam waktu yang sama.

Penjadwalan dibutuhkan untuk mengatur waktu kerja, sehingga jadwal yang efisien bisa didapatkan. Penjadwalan akan terlihat mudah Ketika jumlah komponen yang dijadwalkan sedikit, tetapi menjadi rumit ketika jumlah komponennya banyak [1]. Dengan jumlah kelas dan guru yang banyak, sehingga sulit untuk melakukan kegiatan penjadwalan.

Untuk menyelesaikan permasalahan penjadwalan tersebut, dapat menggunakan algoritma optimasi. Salah satu algoritma optimasi misalnya adalah algoritma Nearest Neighbor, yang dapat digunakan untuk mengotimalkan permasalahan *Vehicle Routing Problem* (VRP) pada kasus transportasi pengangkutan sampahdi Kota padang [2]. Algoritma genetika merupakan salah satu cara untuk memecahkan masalah penjadwalan yang memberikan solusi yang baik. Algoritma genetika menggunakan prinsip seleksi alam dalam ilmu genetika untuk mengembangkan solusi masalah [3]. Solusi ini dievaluasi menggunakan fungsi fitness, diikuti dengan proses seleksi, crossover, dan mutasi. Metode seleksi yang digunakan dalam penelitian ini adalah roulette-wheel dan pindah silang dilakukan dengan crossover satu titik.

Pada penelitian sebelumnya didapatkan hasil berupa algoritma genetika sangat efektif dalam mendesign *time table* untuk sebuah universitas, dan *time table* yang dihasilkan bisa mencapai nilai *fitness* yang lebih baik dibandingkan respon yang dihasilkan melalui metode manual [4]. Algoritma genetika adalah solusi yang cukup baik untuk masalah penjadwalan yang membutuhkan waktu pengerjaan yang lama jika dilakukan secara manual.

2. METODE PENELITIAN

a. Perhitungan Nilai Fitness

Nilai *fitness* mewakili tingkat kebugaran individu. Semakin tinggi nilai *fitness*, semakin tinggi pula tingkat kebugaran dan semakin tinggi kemungkinan untuk lolos seleksi [5]. Nilai *fitness* dihitung berdasarkan parameter berikut:

- 1) Terjadi kemunculan nomor guru yang sama pada kelas berbeda dan dalam jam yang sama.
- 2) Terjadinya kemunculan mata pelajaran lebih dari jam yang telah ditentukan dalam satu paket tatap muka seminggu.

Semakin banyak terdapat kejadian 1 dan 2 pada susunan kromosom, maka nilai *fitness* akan semakin kecil. Proses perhitungan nilai *fitness* sendiri adalah dengan menjumlahkan hasil perhitungan *fitness* masing-masing point lalu dibagi menjadi dua.

b. Seleksi Roulette

Tujuan dari proses seleksi adalah menyaring beberapa individu atau kromosom dengan nilai *fitness* yang optimal untuk dilanjutkan ke proses algoritma genetika selanjutnya. Metode seleksi yang akan digunakan adalah seleksi *roulette*, masing-masing kromosom menempati potongan-potongan lingkaran secara proporsional sesuai dengan nilai *fitness*-nya [6]. Kromosom yang bernilai *fitness* tinggi memiliki kesempatan terpilih lebih keras, namun tidak menutup kemungkinan kromosom yang bernilai *fitness* rendah akan terpilih juga [7]. Proses seleksi *roulette* adalah sebagai berikut:

1) Nilai probabilitas

Nilai probabilitas adalah angka yang diperoleh dari nilai *fitness* masing-masing individu dan akan digunakan untuk menentukan lolos tidaknya dalam seleksi. Semakin tinggi nilai *fitness* maka semakin tinggi peluang lolos seleksi. Rumus untuk menentukan nilai probabilitas untuk setiap individu adalah:

$$\text{Nilai Probabilitas} = \frac{\text{nilai fitness}}{\text{total nilai fitness}} \quad (1)$$

Dengan rumus diatas, maka setiap individu akan memiliki kisaran probabilitas yang berbeda sesuai dengan nilai *fitness* yang mereka miliki.

2) Nilai kumulatif

Setelah menentukan nilai probabilitas untuk setiap individu, selanjutnya dihitung nilai kumulatif berdasarkan nilai probabilitas individu yang telah dibahas sebelumnya. Nilai kumulatif akan berperan sebagai nilai batas atas dalam proses seleksi, rumus yang digunakan dalam perhitungan nilai kumulatif adalah sebagai berikut:

$$K[n] = P[1] + P[2] + P[3] + \dots + P[n] \quad (2)$$

3) Mengacak bilangan 0-1

Setelah menentukan nilai kumulatif, selanjutnya adalah mengacak bilangan antara 0 sampai dengan 1. Hasil angka random akan disesuaikan dengan nilai kumulatif, yang berperan dalam batas atas, hasil seleksi ini akan ditempatkan pada array baru sebagai populasi hasil seleksi.

c. Crossover

Crossover adalah salah satu bagian dari algoritma genetika yang bekerja dengan menyilangkan satu individu dengan individu lainnya [8]. Individu yang memenuhi syarat *crossover* dipotong dua dan akan menghasilkan dua set gen sendiri. Masing-masing set gen ini dipasangkan dengan set gen individu lain. Tujuan dari *crossover* adalah untuk meningkatkan keanekaragaman string dalam suatu populasi dengan menyilangkan antar-string yang diperoleh sebelumnya [9].

1) Crossover rate

Crossover rate adalah angka yang digunakan sebagai gerbang individu apakah persyaratan *crossover* terpenuhi atau tidak. Tidak ada aturan khusus dalam menentukan nilai *crossover rate*. *Crossover* dilakukan ketika nilai *crossover* memenuhi persyaratan. Jika tidak, individu tersebut akan melewati langkah ini.

2) Proses crossover

Hal pertama yang harus dilakukan saat melakukan proses *crossover* adalah membuat pasangan individu. Tidak ada aturan khusus yang mengatur penentuan pasangan ini. Setelah dilakukan pemasangan individu, hal selanjutnya yang harus dilakukan adalah membuat *crossover rate*. Jika memenuhi persyaratan, maka akan dilakukan penyilangan gen. Langkah-langkah *crossover* dijelaskan sebagai berikut:

- Menentukan *crossover rate*, misalnya 0,6. Nilai *crossover rate* menentukan probabilitas individu untuk *crossover*.
- Memasangkan individu dengan tetangganya, dengan urutan 1 dan 2, 3 dan 4, 5 dan 6, 7 dan 8, 9 dan 10. Hingga terdapat 5 pasangan *crossover* dari 10 individu.
- Mengacak angka antara 0 dan 1 untuk memberi nilai *crossover rate* pada setiap pasangan. Jika terdapat nilai yang lebih rendah dari *crossover rate*, maka pasangan individu memenuhi persyaratan *crossover*.
- Ketika pasangan individu memenuhi persyaratan *crossover*, proses *crossover* dilakukan dengan memotong individu pada titik tengah tiap individu, kemudian ditukar antara titik tengah individu 1 dan individu 2 hingga didapat 2 individu baru.

d. Mutasi

Mutasi adalah salah satu proses dari algoritma genetika, dan cara kerjanya adalah mengganti satu atau beberapa gen pada suatu individu dengan gen lain yang susunannya acak. Tidak ada aturan khusus untuk pola urutan genetik dalam proses mutasi ini, proses mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama [10]. Secara umum, mutasi dapat dilakukan dengan mengacak nilai yang lebih rendah dari prioritas mutasi (*mutation rate*) kemudian membalikkan gen yang ada [11].

1) Mutation rate

Mutation rate adalah angka yang digunakan sebagai prasyarat untuk menentukan apakah suatu individu mampu bermutasi. Menentukan nilai *mutation rate* tidak ada kaidah tertentu.

2) Proses mutasi

Proses mutasi hanya dapat dilakukan jika nilai mutasi memenuhi persyaratan *mutation rate*. Langkah-langkah mutasi adalah:

- Menentukan *mutation rate* yang merupakan nilai acak yang menentukan seberapa besar kemungkinan individu untuk bermutasi. Nilai yang ditentukan pada program ini adalah 0,5.
- Jika hasil pengacakan kurang dari 0,5 maka individu tersebut memenuhi syarat untuk mutasi.
- Jika individu memenuhi syarat, proses selanjutnya mengacak angka kembali, jika nilai acak kurang dari 0,5 akan dilakukan mutasi pada *fitness* 1 untuk menukar nomor guru yang bertabrakan pada satu waktu yang sama, dan jika nilai acak lebih dari 0,5 akan dilakukan mutasi pada *fitness* 2, yaitu memperbaiki kelengkapan jadwal pelajaran pada satu kelas dalam satu minggu waktu mengajar.

3. HASIL DAN PEMBAHASAN

Dalam pengujian sistem penjadwalan menggunakan algoritma genetika ada beberapa parameter pengujian yang digunakan, diantaranya adalah jumlah populasi, *crossover rate*, *mutation rate*, dan *epoch* atau generasi yang dapat dilihat sebagai berikut.

a. Hasil pengujian jumlah populasi

Pada bagian ini dijelaskan hasil pengujian jumlah generasi pada sistem penjadwalan dengan perubahan jumlah generasi berupa 10, 20, 30, 40, dan 50. Nilai *crossover rate*, *mutation rate*, dan iterasi yang sudah ditetapkan, yaitu untuk nilai *crossover rate* = 0.5, nilai *mutation rate* = 0.5, dan

nilai iterasi = 10.000. Dari pengujian ini akan didapatkan jumlah generasi yang memiliki nilai *fitness* terbaik.

Tabel 1. Hasil Pengujian Jumlah Populasi

Pengujian ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Maksimal
1	10	0,5	0,5	10.000	0,9127523291925466
2	20	0,5	0,5	10.000	0,9798136645962733
3	30	0,5	0,5	10.000	0,930383022774327
4	40	0,5	0,5	10.000	0,9356884057971013
5	50	0,5	0,5	10.000	0,9580745341614907

Berdasarkan pengujian yang dilakukan, dapat dilihat bahwa perubahan nilai populasi berpengaruh terhadap nilai *fitness* maksimal, dengan menambahkan jumlah populasi nilai *fitness* maksimal yang didapat cenderung meningkat. Pada jumlah populasi 20 merupakan jumlah generasi yang memiliki nilai *fitness* maksimal terbaik, akan tetapi pada pengujian selanjutnya yaitu jumlah populasi 30 terjadi penurunan nilai *fitness* maksimal.

b. Hasil pengujian *crossover rate*

Pada bagian ini dijelaskan hasil pengujian perubahan nilai *crossover rate* pada sistem penjadwalan. Nilai *crossover rate* yang diujikan adalah 0.5, 0.6, 0.7, 0.8, dan 0.9. Dengan nilai *mutation rate* = 0.5, iterasi = 10.000, dan jumlah populasi = 20 berdasarkan jumlah generasi terbaik dari pengujian yang dilakukan pada subbab sebelumnya.

Tabel 2. Hasil Pengujian Crossover Rate

Pengujian ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Maksimal
1	20	0,1	0,5	10.000	0,8856592908902692
2	20	0,2	0,5	10.000	0,8902691511387164
3	20	0,3	0,5	10.000	0,9532705745341614
4	20	0,4	0,5	10.000	0,9022386128364389
5	20	0,5	0,5	10.000	0,9704968944099379
6	20	0,6	0,5	10.000	0,984472049689441
7	20	0,7	0,5	10.000	0,9767080745341614
8	20	0,8	0,5	10.000	0,9387939958592133
9	20	0,9	0,5	10.000	0,9798136645962733

Berdasarkan pengujian yang dilakukan dapat dilihat bahwa perubahan *crossover rate* berpengaruh pada nilai *fitness* maksimal, nilai *crossover rate* 0.6 menghasilkan nilai *fitness* maksimal terbaik. Tetapi pada pengujian ke 3, 4, dan 5 menunjukkan penurunan nilai *fitness* terbaik, hal ini disebabkan semakin tingginya nilai *crossover rate*, maka akan semakin banyak kromosom yang mengalami *crossover*, semakin banyak kromosom yang *ter-crossover*, maka akan dapat merusak kromosom yang sudah memiliki nilai *fitness* yang tinggi.

c. Hasil pengujian *mutation rate*

Pada bagian ini dijelaskan hasil pengujian perubahan *mutation rate* pada sistem penjadwalan. Proses mutasi dilakukan untuk memperbaiki jadwal yang belum sesuai, pada proses mutasi berperan dalam memperbaiki guru yang bertabrakan dalam satu waktu yang sama serta memperbaiki distribusi pelajaran yang belum lengkap. Nilai *mutation rate* yang akan diujikan adalah 0.1, 0.2, 0.3, 0.4, dan 0.5. Dengan nilai iterasi yang telah ditetapkan yaitu 10.000, sedangkan jumlah populasi dan *crossover rate* berdasarkan jumlah generasi dan *crossover rate* terbaik yang didapatkan dari pengujian pada subbab sebelumnya, yaitu jumlah populasi = 20 dan *crossover rate* = 0.6.

Tabel 3. Hasil Pengujian Mutation Rate

Pengujian ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Maksimal
1	20	0,6	0,1	10.000	0,937597049689441
2	20	0,6	0,2	10.000	0,9891304347826086
3	20	0,6	0,3	10.000	0,9906832298136646
4	20	0,6	0,4	10.000	0,9543219461697723
5	20	0,6	0,5	10.000	0,9704968944099379
6	20	0,6	0,6	10.000	0,9527691511387163
7	20	0,6	0,7	10.000	0,9565217391304348
8	20	0,6	0,8	10.000	0,9350414078674948
9	20	0,6	0,9	10.000	0,8965773809523809

Berdasarkan pengujian yang dilakukan dapat meningkatkan nilai *fitness* maksimal, tetapi apabila mutation rate yang digunakan terlalu besar maka nilai *fitness* terbaik akan hilang sehingga mengakibatkan turunnya nilai *fitness* maksimal, nilai *mutation rate* terbaik adalah 0.3 berdasarkan dari nilai *fitness* maksimal terbaik yang didapat.

d. Hasil pengujian jumlah iterasi

Pada bagian ini dijelaskan hasil pengujian perubahan jumlah iterasi pada sistem penjadwalan, yaitu bernilai 10.000, 25.000, 50.000, 75.000, dan 100.000. Dengan nilai populasi = 20, *crossover rate* = 0,6, dan *mutation rate* = 0,3 berdasarkan hasil pengujian pada subbab sebelumnya.

Tabel 4. Hasil Pengujian Jumlah Iterasi

Pengujian ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Maksimal
1	20	0,6	0,3	1.000	0,9767080745341614
2	20	0,6	0,3	2.000	0,9496635610766047
3	20	0,6	0,3	3.000	0,9276818064182195
4	20	0,6	0,3	4.000	0,9372412008281573
5	20	0,6	0,3	5.000	0,9798136645962733
6	20	0,6	0,3	6.000	0,9527691511387164
7	20	0,6	0,3	7.000	0,9043898809523809
8	20	0,6	0,3	8.000	0,9860248447204969
9	20	0,6	0,3	9.000	0,9891304347826086
10	20	0,6	0,3	10.000	0,9751552795031055

Berdasarkan pengujian yang dilakukan, dapat disimpulkan peningkatan nilai iterasi berpengaruh terhadap peningkatan nilai *fitness* maksimal, jumlah iterasi 9.000 merupakan jumlah iterasi yang menghasilkan nilai *fitness* maksimal terbaik. Berdasarkan pengujian yang dilakukan, dapat disimpulkan peningkatan nilai iterasi berpengaruh terhadap peningkatan nilai *fitness* maksimal, jumlah iterasi 9.000 merupakan jumlah iterasi yang menghasilkan nilai *fitness* maksimal terbaik.

e. Hasil pengujian akhir

Berdasarkan nilai jumlah populasi, *crossover rate*, *mutation rate*, dan juga iterasi yang didapat pada subbab sebelumnya, dilakukan 10 kali pengujian seperti yang bisa dilihat pada tabel 5.

Tabel 5. Hasil Pengujian Akhir

Uji ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Terbaik	Run Time
1	20	0,6	0,3	9.000	0.9704968944 099379	10,5671879 detik

Uji ke-	Populasi	Crossover Rate	Mutation Rate	Iterasi	Nilai <i>Fitness</i> Terbaik	Run Time
2	20	0,6	0,3	9.000	0,9751552795 031055	13,5822783 detik
3	20	0,6	0,3	9.000	0,9937888198 757764	13,0292414 detik
4	20	0,6	0,3	9.000	0,9767080745 341614	12,7279062 detik
5	20	0,6	0,3	9.000	0,9844720496 89441	12,7896152 detik
6	20	0,6	0,3	9.000	0,9860248447 204969	15,8437904 detik
7	20	0,6	0,3	9.000	0,9402012163 561076	12,2675264 detik
8	20	0,6	0,3	9.000	0,9860248447 204969	12,1511574 detik
9	20	0,6	0,3	9.000	0,9704968944 099379	14,9875618 detik
10	20	0,6	0,3	9.000	0,9512163561 076605	12,6332167 detik

4. KESIMPULAN

Berdasarkan dari hasil pengujian yang dilakukan memberikan hasil terbaik pada pengujian ke 3 yaitu dengan nilai *fitness* terbaik 0,99 atau hampir mendekati 1. Nilai *fitness* terbaik didapat dari pemilihan kromosom dengan nilai tabrakan antar guru dan pelajaran yang sedikit serta jumlah pelajaran yang terdistribusi secara merata, hasil pengujian membuktikan pembuatan jadwal menggunakan algoritma genetika menghasilkan jadwal pelajaran yang hampir sempurna dengan waktu yang diperlukan sangat singkat, yaitu 13 detik.

DAFTAR PUSTAKA

- [1] Kurnia, N., Wayan, M., & Mahmudy, F. (2006). Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika. 2(2), 1–8.
- [2] Susanti, D. (2018). Analisis Jarak pada Rute Truk Pengangkutan Sampah Dengan Metode Vehicle Routing Problem (VRP) Kota Padang. Teknomatika, 8(1).
- [3] Hidayat, I., Revo, S., Inkiriwang, L., & Pratasis, P. A. K. (2019). Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika Pada Proyek Rehabilitasi Puskesmas Minanga. Jurnal Sipil Statik, 7(12), 1669–1680.
- [4] Abbaszadeh, M., & Saeedvand, S. (2014). A Fast Genetic Algorithm for Solving University Scheduling Problem. IAES International Journal of Artificial Intelligence (IJ-AI), 3(1), 7. <https://doi.org/10.11591/ijai.v3.i1.pp7-15>
- [5] Maharsi, A. L. (2013). Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika
- [6] Elva, Y. (2019). Sistem Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika. Jurnal Teknologi Informasi, 3(1).
- [7] Amik, L. T., Gama, M., Kayangan, J., 99, N., & Riau, D.-. (2017). Implementasi Algoritma Genetika dalam Pembuatan Jadwal Kuliah. In Jaringan Sistem Informasi Robotik (Vol. 1, Issue 01)
- [8] Mauluddin, S., Ikbali, I., & Nursikuwagus, A. (2018). Optimasi aplikasi penjadwalan kuliah menggunakan algoritma genetika. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2(3), 792-799.

- [9] Assagaf, A., Ibrahim, A., & Suranto, C. (2018). Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara. 2621–4962.
 - [10] Suprayogi, D. A., & Mahmudy, W. F. (2015). Penerapan algoritma genetika traveling salesman problem with time window: Studi kasus rute antar jemput laundry. *Jurnal Buana Informatika*, 6(2).
 - [11] Luh, N., Sri, W., Ginantra, R., Bagus, I., & Anandita, G. (2019). Implementasi Algoritma Genetika Berbasis Web pada Sistem Penjadwalan Mengajar di SMK Dwijendra Denpasar. *Jurnal Teknologi Informasi Dan Komputer*.
-