Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 5 No. 3 (2021) 550 - 556 ISSN Media Elektronik: 2580-0760

Model Paralelisasi Algoritma Genetika Terpandu pada Sistem Penjadwalan Kuliah Universitas dengan Alokasi Waktu Dinamis

Muhammad Fachrie[1], Anita Fira Waluyo[2]
[1,2]Program Studi Informatika, Fakultas Sains & Teknologi
Universitas Teknologi Yogyakarta
[1]muhammad.fachrie@staff.uty.ac.id, [2]anitafira@staff.uty.ac.id

Abstract

One of the many techniques used to solve the University Course Timetable Problem (UCTP) is Genetic Algorithm (GA) which is a technique in the field of Evolutionary Computation. However, GA has high computational complexity due to the large number of evolutionary operators that must be performed during the evolutionary process, so it takes a long time to produce an optimal timetable. The computation time will also increase when the number of optimized variables is very large, such as in UCTP. Of course, this makes the application less reliable by users. Therefore, this article proposes a parallelization model for GA to reduce computation time in solving UCTP problems. The proposed AG is designed with a multithreading CPU scheme and implements a guided creep mutation mechanism and eliminates the recombination mechanism to reduce more computation time. The proposed system was tested and evaluated using two different UCTP datasets from the University of Technology Yogyakarta which contained 878 and 1140 lecture meetings in even and odd semesters. Unlike the previous ones, this study discusses UCTP with dynamic time slots where the duration of the lecture depends on the course credits. From the tests that have been done, it is found that the GA that was built is able to generate optimal course timetable without any clashes in a relatively fast time, that is less than 60 minutes for 1140 lecture meetings and less than 20 minutes for 878 lecture meetings. The use of the multithreading CPU model has succeeded in reducing computation time by 62% when compared to the conventional model which only uses one thread.

Keywords: evolutionary computation, genetic algorithm, optimization, parallelization, university course timetable

Abstrak

Salah satu teknik yang banyak digunakan untuk menyelesaikan *University Course Timetable Problem* (UCTP) adalah Algoritma Genetika (AG) yang merupakan salah satu teknik di dalam bidang Komputasi Evolusioner. Namun, AG memiliki kompleksitas komputasi yang tinggi karena banyaknya operator evolusioner yang harus dilakukan selama periode evolusi, sehingga membutuhkan waktu yang lama untuk menghasilkan jadwal perkuliahan yang optimal. Waktu komputasi pun akan semakin meningkat ketika jumlah variabel yang dioptimasi berjumlah sangat banyak seperti pada UCTP. Tentunya hal tersebut menjadikan aplikasi kurang dapat diandalkan oleh pengguna. Oleh karena itu, artikel ini mengusulkan model paralelisasi pada AG untuk mengurangi waktu komputasi dalam menyelesaikan permasalahan UCTP. AG yang diusulkan ini dirancang dengan skema CPU *multithreading* dan menerapkan mekanisme *guided creep mutation* serta menghilangkan mekanisme rekombinasi untuk mengurangi lebih banyak waktu komputasi. Sistem yang diusulkan diuji dan dievaluasi dengan menggunakan dua dataset UCTP yang berbeda dari Universitas Teknologi Yogyakarta yang berisi 878 dan 1140 pertemuan perkuliahan pada semester genap dan ganjil. Berbeda dengan sebelumnya, penelitian ini membahas UCTP dengan slot waktu dinamis dimana durasi perkuliahan bergantung pada SKS mata kuliah. Dari pengujian yang telah dilakukan, didapatkan hasil bahwa AG yang dibangun mampu menyusun jadwal perkuliahan dengan optimal tanpa adanya bentrok dalam waktu yang relatif cepat, yakni kurang dari 60 menit untuk 1140 pertemuan perkuliahan dan kurang dari 20 menit untuk 878 pertemuan perkuliahan. Penggunaan model CPU *multithreading* berhasil mengurangi waktu komputasi hingga 62% jika dibandingkan dengan model konvensional yang hanya menggunakan satu *thread* saja.

Kata kunci: algoritma genetika, komputasi evolusioner, optimasi, paralelisasi, penjadwalan kuliah universitas

1. Pendahuluan

Dalam bidang optimasi kombinatorial, penjadwalan kuliah di tingkat universitas atau yang dikenal dengan istilah *University Course Timetable Problem* (UCTP) merupakan salah satu objek penelitian yang memiliki kompleksitas yang sangat tinggi karena memiliki ruang

Diterima Redaksi: 28-02-2021 | Selesai Revisi: 19-06-2021 | Diterbitkan Online: 26-06-2021

pencarian yang sangat luas, sehingga UCTP digolongkan ke dalam permasalahan Polinomial Non Deterministik (*NP-hard problem*) [1]–[3]. Selain itu, UCTP memiliki berbagai macam variasi bergantung pada tingkat pendidikan, aturan di dalam sekolah atau perguruan tinggi, bahkan kebijakan pemerintah dalam suatu negara [4]. Oleh karena itu, pendekatan berbasis probabilistik menjadi pilihan yang tepat untuk menyelesaikan permasalahan dengan ruang pencarian yang sangat luas tersebut. Salah satu metode yang banyak dipakai adalah Algoritma Genetika (AG) yang merupakan salah satu algoritma di dalam bidang *Evolutionary Computation* (EC) yang dikenal sangat andal dalam menyelesaikan permasalahan kompleks seperti UCTP [5].

AG memiliki kompleksitas waktu yang tinggi, sebab terdapat sejumlah mekanisme evolusi yang harus dieksekusi, di antaranya, seleksi kromosom orang tua, rekombinasi, hingga mutasi kromosom. Dari serangkaian mekanisme tersebut, rekombinasi adalah bagian yang cukup menguras waktu, sebab mekanisme ini harus didahului oleh proses seleksi kromosom orang tua. Waktu komputasi akan semakin meningkat seiring dengan bertambahnya jumlah variabel yang dioptimasi. Sebagai gambaran, pada UCTP, jumlah variabel yang harus dioptimasi berjumlah ratusan bahkan ribuan variabel. Penelitian terbaru yang dipublikasikan pada [6], [7] mengusulkan AG dengan *Island Model*, yakni sebuah model paralelisasi AG dengan mendistribusikan proses evolusi ke sejumlah komputer yang saling terhubung. Pada model tersebut, satu komputer bertindak sebagai *master* untuk mengalokasikan dan menghimpun hasil pemrosesan dari komputer-komputer lain (*slave*) yang ada di bawah kendalinya. *Island Model* tersebut sangat cocok untuk menyelesaikan UCTP dengan jumlah perkuliahan yang sangat besar dan memiliki mekanisme pengecekan bentrok hingga level mahasiswa. Meskipun begitu, model ini membutuhkan setidaknya tiga buah komputer untuk dapat memberikan kinerja yang baik. Pada [8] dan [9], model yang serupa juga pernah diusulkan namun dengan arsitektur AG yang lebih sederhana.

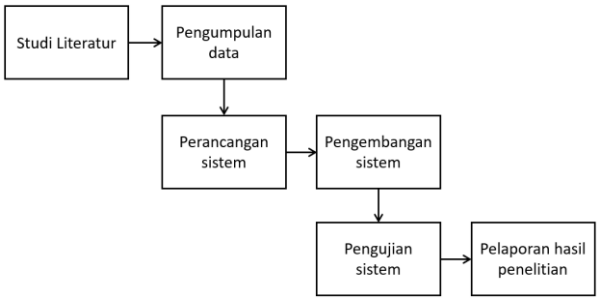
Selain itu, pada kondisi tertentu, AG seringkali terjebak pada solusi optimum lokal selama proses evolusi [10]. Kondisi ini mengakibatkan AG sangat lambat (membutuhkan banyak generasi) atau sama sekali tidak dapat menemukan solusi terbaik atau yang disebut dengan optimum global. Pendekatan yang banyak digunakan untuk mengatasi masalah tersebut adalah dengan menggunakan strategi pencarian lokal (*local search*) [3], [11]–[15]. Selain itu, pendekatan lainnya yang juga pernah diusulkan adalah pencarian terpandu (*guided search*) di mana kromosom akan direkombinasi atau dimutasi berdasarkan suatu aturan tertentu sehingga kromosom yang dihasilkan mampu memenuhi aturan-aturan penjadwalan yang ditentukan oleh sekolah atau perguruan tinggi [3], [5], [16], [17].

Adapun pada penelitian ini, diusulkan model AG paralel dengan skema *CPU multithreading*. Skema ini digunakan untuk menyelesaikan UCTP dengan jumlah perkuliahan yang lebih sedikit daripada [6], [7] dan mekanisme pengecekan bentrok yang tidak sampai pada level mahasiswa. Penggunaan model paralel tersebut dibutuhkan, mengingat jumlah perkuliahan yang harus dijadwalkan sangat banyak, yakni mencapai lebih dari seribu perkuliahan, sehingga skema pemrograman AG harus dapat mengoptimalkan sumber daya CPU untuk dapat bekerja secara paralel. Di samping itu, agar semakin meningkatkan efisiensi dan efektifitas sistem penjadwalan, maka model AG yang dibangun menerapkan skema *guided search* berupa mutasi *creep* berbasis aturan (*guided creep mutation*). Model AG tersebut tidak melibatkan operator rekombinasi dan seleksi kromosom pada proses evolusinya sehingga proses evolusi lebih efisien namun tetap efektif dalam menemukan solusi optimum global [5]–[7].

2. Metode Penelitian

2.1. Tahapan Penelitian

Penelitian ini bersifat eksperimental yang didukung dengan data-data riil yang ada di lapangan. Sebagaimana yang disajikan pada Gambar 1, tahap awal penelitian diawali dengan aktivitas studi literatur dari berbagai artikel jurnal maupun prosiding dari dalam dan luar negeri, serta mengumpulkan sejumlah data yang meliputi data jadwal perkuliahan dari semester ganjil dan genap yang sudah berlalu, data ruang kelas dan laboratorium praktikum, serta kebijakan-kebijakan terkait penjadwalan kuliah di Universitas Teknologi Yogyakarta (UTY). Kemudian, data jadwal kuliah yang sudah diperoleh dirapikan dan dianalisis untuk mempelajari pola penjadwalan yang biasa dilakukan oleh Bagian Operasional (BO).



Gambar 1. Alur penelitian yang dilakukan dalam pengembangan model paralelisasi pada Algoritma Genetika untuk penjadwalan perkuliahan tingkat universitas.

Setelah data jadwal kuliah siap digunakan, maka perancangan aplikasi pun dilakukan dengan memperhatikan aturan-aturan yang berlaku. Sayangnya, aturan-aturan tersebut tidak tertulis secara legal formal, melainkan hanya menjadi strategi penyusunan jadwal yang digunakan oleh petugas penjadwalan kuliah. Proses perancangan aplikasi ini menghasilkan beberapa

DOI: https://doi.org/10.29207/resti.v5i3.2988

keluaran, di antaranya rancangan *Graphical User Interface* (GUI), struktur data, arsitektur dan parameter Algoritma Genetika, serta rancangan awal struktur kode program.

Aplikasi dibangun menggunakan Bahasa Pemrograman Java berbasis *desktop*. Aplikasi yang telah dibangun melalui beberapa tahap pengujian, yakni pengujian fungsionalitas umum aplikasi yang meliputi input dan output, lalu dilanjutkan dengan pengujian awal Algoritma Genetika untuk menjadwalkan perkuliahan, kemudian dilanjutkan lagi dengan pengujian Algoritma Genetika untuk penjadwalan kuliah dengan beberapa skenario pengujian. Semua hasil pengujian dicatat dan sejumlah data penting disajikan di dalam publikasi.

2.2. Model Penjadwalan Kuliah

Pada penelitian ini, model penjadwalan kuliah mengacu pada aturan dan kebijakan yang digunakan pada Universitas Teknologi Yogyakarta (UTY). Secara umum, perkuliahan di UTY memiliki beberapa karakteristik, di antaranya, beban mengajar masing-masing dosen tergolong tinggi dengan rata-rata beban mengajar sebanyak 18 hingga 24 SKS tiap semesternya, alokasi waktu pada jadwal kuliah bersifat dinamis karena durasi perkuliahan pada setiap sesi bergantung pada bobot mata kuliah (SKS), selain itu, dosen-dosen yang memiliki tugas tambahan sebagai pejabat struktural memiliki jadwal kuliah khusus agar tidak mengganggu agenda rapat struktural.

Tingginya beban mengajar tiap dosen mengharuskan adanya strategi yang tepat agar perkuliahan yang diampu tetap terlaksana dengan efektif dan dosen pun tetap merasa nyaman dengan jadwal perkuliahan yang diberikan. Oleh karena itu, beban mengajar dosen dibatasi maksimal 6 SKS setiap harinya agar beban mengajar terdistribusi ke hari lainnya dan tidak menumpuk pada hari-hari tertentu saja. Dosen pun sebaiknya tidak berada di kampus lebih dari 7 jam sehari, maka jadwal kuliah yang dibuat pun tidak boleh terpaut terlalu jauh antara satu sesi dengan sesi berikutnya. Apabila dosen mengajar hingga sesi terakhir pada suatu hari, maka keesokan harinya dosen tersebut tidak diperkenankan mengajar pada sesi pertama untuk memberikan kenyamanan dan istirahat yang cukup bagi dosen tersebut.

Sebagaimana terlihat pada Gambar 2, model penjadwalan kuliah dengan alokasi waktu dinamis sangat berbeda dengan model alokasi waktu statis yang banyak digunakan selama ini. Model penjadwalan dengan alokasi waktu dinamis tersebut bertujuan agar setiap mata kuliah mendapatkan durasi pembelajaran yang tepat sesuai dengan beban mata kuliah (SKS) tersebut. Semakin besar SKS mata kuliah tersebut, maka semakin lama pula alokasi waktu perkuliahan yang diberikan. Pada alokasi waktu dinamis, jumlah sesi pada satu hari dan satu ruang bisa berbeda dengan hari dan ruang lainnya.

Senin	Selasa	Rabu	Kamis	Jumat
2 jam	2 jam	2 jam	2 jam	2 jam
2 jam	2 jam	2 jam	2 jam	2 jam
2 jam	2 jam	2 jam	2 jam	2 jam
2 jam	2 jam	2 jam	2 jam	2 jam

Senin	Selasa	Rabu	Kamis	Jumat
3 jam	2 jam	4 jam	3 jam	2 jam
	2 jam			3 jam
2 jam			3 jam	
	2 jam	4 jam		
3 jam				3 jam
	2 jam		2 jam	

Gambar 2. Ilustrasi perbedaan antara jadwal perkuliahan dengan alokasi waktu statis dan alokasi waktu dinamis.

Berdasarkan karakteristik perkuliahan yang telah dijabarkan, serta didasarkan pada kebijakan universitas, maka diperoleh sejumlah regulasi atau batasan (*constraint*) yang digunakan sebagai acuan dalam menyusun jadwal perkuliahan. Regulasi atau batasan tersebut terbagi menjadi dua jenis, yakni batasan wajib atau yang sering dikenal dengan istilah *hard constraint* (HC) dan batasan lunak atau yang biasa disebut dengan *soft constraint* (SC).

HC merupakan batasan yang sama sekali tidak boleh dilanggar oleh suatu jadwal kuliah. Pelanggaran terhadap HC mengakibatkan jadwal kuliah yang dihasilkan tidak valid. Sedangkan SC merupakan batasan yang sifatnya lebih luwes namun sebaiknya tidak dilanggar. Apabila ada aturan pada SC yang dilanggar, maka jadwal kuliah yang dihasilkan tetap valid meskipun tidak sepenuhnya sesuai dengan ekspektasi yang diharapkan. Tabel 1 menyajikan himpunan HC dan SC yang pada penelitian ini.

Tabel 1. Himpunan batasan wajib (HC) dan batasan lunak (SC)

Batasan	Tipe	Penalti
Tidak ada bentrok dosen	HC1	100
Tidak ada bentrok kelas mahasiswa	HC2	100
Tidak ada bentrok ruang	HC3	100
Praktikum harus dilaksanakan di laboratorium	HC4	100
Perkuliahan harus dilaksanakan pada ruangan dengan kapasitas yang cukup	HC5	100
Dosen dengan jabatan struktural tidak boleh mengajar di bawah pukul 10.30	HC6	100
Dosen sebaiknya tidak mengajar lebih dari 6 SKS dalam sehari	SC1	10
Dosen sebaiknya tidak mengajar di atas pukul 12.00 pada Hari Sabtu	SC2	10
Dosen sebaiknya tidak mengajar pada sesi pertama jika sehari sebelumnya ia mengajar pada sesi terakhir	SC3	10
Dosen sebaiknya tidak menunggu di kampus lebih dari 7 jam sehari	SC4	10
Jumlah perkuliahan pada Hari Sabtu sebaiknya jauh lebih sedikit dari hari lainnya	SC5	10
Mahasiswa sebaiknya tidak mengikuti perkuliahan lebih dari 6 SKS dalam sehari	SC6	5

Jadwal perkuliahan yang optimum adalah jadwal yang memenuhi semua batasan wajib (HC) dan memenuhi sebanyak mungkin batasan lunak (SC). Artinya, sistem harus menghilangkan pelanggaran pada HC dan meminimalisasi jumlah pelanggaran pada SC. Setiap

DOI: https://doi.org/10.29207/resti.v5i3.2988

pelanggaran terhadap HC maupun SC akan diberikan penalti sebagaimana yang ditunjukkan pada Tabel 1.

Spesifikasi ruang dan waktu juga diperlukan untuk mengembangkan sistem penjadwalan kuliah tersebut. Tabel 2 dan Tabel 3 menyajikan data spesifikasi ruang dan juga waktu yang digunakan pada penelitian ini, di mana spesifikasi ini didasarkan pada kondisi yang ada di Universitas Teknologi Yogyakarta (UTY).

Tabel 2. Spesifikasi ruangan perkuliahan

Ruangan	Spesifikasi
Ruang kelas ukuran S	1 ruangan (maks. 40 mahasiswa)
Ruang kelas ukuran M	15 ruangan (maks. 60 mahasiswa)
Ruang kelas ukuran L	32 ruangan (maks. 70 mahasiswa)
Ruang kelas ukuran XL	5 ruangan (maks. 90 mahasiswa)
Laboratorium ukuran S	11 ruangan (maks. 45 mahasiswa)
Laboratorium ukuran L	2 ruangan (maks. 70 mahasiswa)

Tabel 3. Spesifikasi waktu mulai perkuliahan

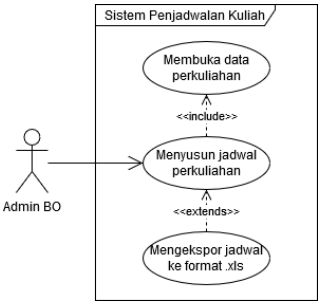
Waktu mulai	Keterangan
07.00	Cocok untuk semua mata kuliah
08.50	Cocok untuk semua mata kuliah
09.40	Cocok untuk semua mata kuliah
10.40	Cocok untuk semua mata kuliah
12.50	Cocok untuk semua mata kuliah
14.40	Cocok untuk semua mata kuliah
15.30	Tidak cocok untuk mata kuliah dengan SKS > 3
16.30	Tidak cocok untuk mata kuliah dengan SKS > 2

Pada sistem penjadwalan yang dikembangkan, waktu akhir dari tiap sesi tidak didefinisikan di awal, melainkan ditentukan secara otomatis saat program dieksekusi berdasarkan bobot SKS dari mata kuliahnya, di mana 1 SKS setara dengan 50 menit perkuliahan. Hal ini dilakukan untuk mengakomodasi model alokasi waktu dinamis pada sistem penjadwalan tersebut.

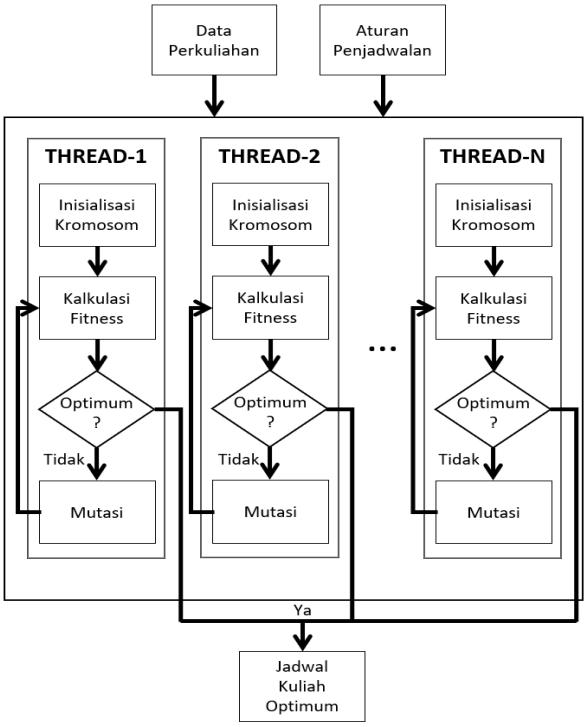
2.3. Arsitektur Sistem

Secara umum, sistem yang dikembangkan memiliki beberapa fitur utama, yakni membuka data perkuliahan yang akan disusun jadwalnya, melakukan penjadwalan otomatis, dan mengekspor jadwal perkuliahan dalam format .xls. Ketiga fitur ini hanya dapat dioperasikan oleh operator yang ditunjuk oleh universitas, dalam hal ini adalah administrator Bagian Operasional (BO) Universitas Teknologi Yogyakarta (UTY) seperti yang terlihat pada Gambar 3. Sistem dikembangkan pada platform *desktop* dengan Bahasa Pemrograman Java. Sistem penjadwalan perkuliahan tersebut ditujukan untuk membantu admin BO menyusun jadwal perkuliahan dan selanjutnya admin BO dapat melakukan penyesuaian atau *editing* terhadap jadwal yang dihasilkan oleh sistem jika diperlukan.

Secara lebih rinci lagi, proses penyusunan jadwal perkuliahan yang melibatkan Algoritma Genetika Terpandu terdiri dari tiga proses utama sebagaimana yang terlihat pada Gambar 4, yakni proses inisialisasi kromosom, perhitungan fitness, dan mutasi kromosom. Ketiga rangkaian proses tersebut dibungkus sebagai satu proses evolusi, kemudian masing-masing proses evolusi tersebut dieksekusi oleh sejumlah *thread* yang terpisah. Satu *thread* dapat memiliki satu atau beberapa kromosom. Ketiadaan operator rekombinasi dan seleksi kromosom orang tua menjadikan sistem tersebut tidak memerlukan interaksi antar kromosom, sehingga masing-masing proses evolusi dalam setiap *thread* dapat berjalan secara independen sepenuhnya. Ketika ada satu *thread* yang telah mencapai nilai fitness tertinggi atau nilai fitness yang diharapkan, maka eksekusi *thread* akan dihentikan dan program akan mengembalikan jadwal terbaik hasil optimasi.



Gambar 3. Use Case Diagram dari sistem penjadwalan kuliah yang dibangun



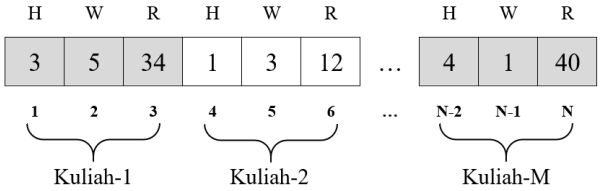
Gambar 4. Model paralelisasi pada sistem penjadwalan berbasis Algoritma Genetika Terpandu yang dikembangkan

Jumlah *thread* yang digunakan tentunya disesuaikan dengan kemampuan *Central Processing Unit* (CPU). Di dalam penelitian ini, komputer yang digunakan ditenagai oleh CPU Intel Core i5 seri 8250u yang memiliki 4 *core* dan 8 *thread*, sehingga pada eskperimen yang dilakukan menerapkan beberapa skenario

DOI: https://doi.org/10.29207/resti.v5i3.2988

multithreading untuk melihat kinerja sistem secara komprehensif. Penggunaan model *multithreading* diharapkan dapat memaksimalkan utilisasi CPU pada proses komputasi.

Proses inisialisasi kromosom dilakukan dengan membangkitkan nilai tiap gen secara acak dalam rentang nilai tertentu. Kromosom dirancang dengan mendekomposisi jadwal dari setiap perkuliahan, yakni dengan mengodekan setiap pertemuan perkuliahan ke dalam tiga gen, yakni gen 'hari' (H), gen 'waktu' (W), dan gen 'ruangan' (R). Sehingga jika jumlah perkuliahan adalah sebanyak M, maka jumlah gen/ panjang kromosom adalah N = 3M. Gambar 5 mendeskripsikan desain kromosom yang digunakan.



Gambar 5. Desain kromosom dengan mendekomposisi variabel perkuliahan ke dalam gen 'hari' (H), 'waktu' (W), dan 'ruang' (R)

Gen H memiliki rentang nilai [1,6] untuk merepresentasikan perkuliahan dari Hari Senin hingga Sabtu (6 hari kerja). Sedangkan gen W berada dalam rentang [1,8] untuk merepresentasikan delapan alokasi waktu perkuliahan yang tersedia dan gen R memiliki rentang nilai [1,66] karena terdapat 66 ruangan yang berbeda.

Adapun fungsi fitness yang digunakan untuk mengevaluasi setiap kromosom selama proses evolusi dapat dilihat pada persamaan 1, di mana variabel *M* adalah jumlah perkuliahan yang dijadwalkan, dan *P* adalah jumlah penalti yang dimiliki oleh kromosom dari semua HC dan SC yang dilanggar. Fungsi fitness ini akan bernilai semakin tinggi apabila semakin kecil pelanggaran yang dimiliki oleh kromosom, begitu pula sebaliknya.

$$fitness = 100 \times M - \left(\sum_{i=1}^{M} P_i\right) \quad (1)$$

Pada sistem yang dibangun, mekanisme evolusi hanya mengandalkan proses mutasi dengan model mutasi *creep* terarah (*guided creep mutation*) yakni mekanisme penggantian nilai gen dengan nilai terdekatnya (berubah perlahan) berdasarkan aturan mutasi yang telah didefinisikan sebagai berikut:

```
Mutasi Creep Terarah
input: sebuah kromosom
for setiap kuliah pada kromosom do:
   ganti_hari, ganti_waktu, ganti_ruang = false

   if kuliah melanggar HC1 or HC2 or SC4:
     ganti waktu or hari
   end if

   if kuliah melanggar HC3:
     ganti waktu or hari or ruang
   end if

   if kuliah melanggar HC4 or HC5:
     if not ganti_ruang:
       ganti ruang
     end if
   end if

   if kuliah melanggar SC1 or S2 or SC5 or SC6:
     if not ganti_hari:
       ganti hari
     end if
   end if

   if kuliah melanggar SC3:
     if not ganti_waktu:
       ganti waktu
     end if
   end if
end if
```

2.4. Dataset

Kinerja sistem yang dibangun diuji dan dievaluasi menggunakan dua buah dataset jadwal kuliah yang diperoleh dari kampus 1 Universitas Teknologi Yogyakarta (UTY). Sebanyak 878 dan 1140 data perkuliahan dari semester genap dan semester ganjil berisi nama mata kuliah, nama dosen pengampu, kelas mahasiswa, jumlah mahasiswa tiap kuliah, SKS mata kuliah, dan jenis mata kuliahnya. Kedua data tersebut merupakan data perkuliahan dari dua fakultas dan 13 program studi. Tabel 4 menyajikan secara rinci mengenai dataset yang digunakan dalam penelitian.

Tabel 4. Rincian dataset yang digunakan dalam penelitian

Semester Genap	Semester Ganjil
1140 perkuliahan	878 perkuliahan
431 mata kuliah	296 mata kuliah
188 dosen	157 dosen
912 mata kuliah teori	660 mata kuliah teori
228 mata kuliah praktikum	218 mata kuliah praktikum

3. Hasil dan Pembahasan

Serangkaian pengujian dengan beberapa skenario telah berhasil dilakukan untuk mengevaluasi kinerja sistem yang dikembangkan. Algoritma Genetika yang diuji pada sistem ini menggunakan populasi sebanyak 10 kromosom dengan probabilitas mutasi sama dengan 1 yang artinya bahwa semua gen yang melanggar HC maupun SC akan dimutasi.

Berdasarkan Tabel 5, terdapat empat skenario yang digunakan untuk menguji model *multithreading* pada sistem penjadwalan kuliah yang dikembangkan. Skenario pertama terdiri dari hanya satu *thread* saja (tanpa *multithreading*) untuk memproses 10 kromosom. Sedangkan pada skenario kedua, ketiga, dan keempat digunakan skema *multithreading* dengan jumlah *thread* 2, 5, dan 10.

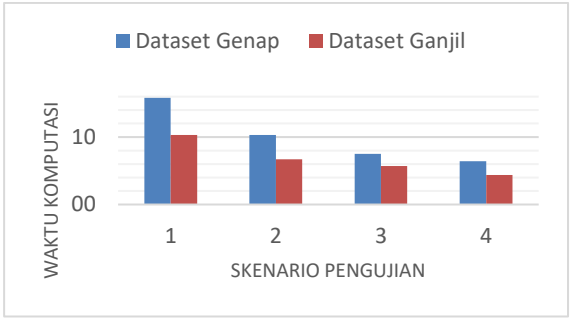
Hasil pengujian pada Gambar 6 menunjukkan adanya perbaikan waktu komputasi pada skenario yang

DOI: https://doi.org/10.29207/resti.v5i3.2988

menerapkan skema *multithreading*. Waktu komputasi yang dibutuhkan untuk memproses satu generasi (iterasi) di dalam proses evolusi menurun seiring dengan bertambahnya jumlah *thread* yang digunakan. Pada skenario ke-2 yang menggunakan dua buah thread yang masing-masingnya memproses lima buah kromosom, terjadi penurunan waktu komputasi hingga mendekati 40% dibandingkan dengan skenario pertama. Pada skenario ketiga di mana terdapat lima buah *thread* yang masing-masingnya mengeksekusi dua buah kromosom, tingkat penurunannya mencapai 56%. Sedangkan penurunan terbesar terjadi pada skenario keempat dengan waktu komputasi hingga 62% lebih singkat daripada skenario ke-1 yang hanya menggunakan satu buah *thread*. Meskipun begitu, jika dilihat lebih jeli lagi pada grafik yang disajikan pada Gambar 6, persentase penurunan waktu komputasi dari skenario ke-2 ke skenario ke-3 dan skenario ke-4 tidak sebesar penurunan waktu komputasi yang terjadi dari skenario ke-1 menuju skenario ke-2. Hal ini dimungkinkan karena pertambahan *thread* yang lebih banyak mengakibatkan adanya antrean proses yang menunggu untuk dieksekusi dikarenakan sebagian besar sumber daya CPU terpakai.

Tabel 5. Skenario *multithreading* yang digunakan dalam penelitian

Skenario	Jumlah *Thread*	Jumlah Kromosom	Jumlah Kromosom Per *Thread*
1	1 *thread*	10	10
2	2 *thread*	10	5
3	5 *thread*	10	2
4	10 *thread*	10	1



Gambar 6. Waktu komputasi rata-rata untuk satu iterasi (generasi) yang didapatkan dari hasil pengujian dengan empat skenario

Pemanfaatan skema *multithreading* pada sistem yang dikembangkan juga meningkatkan penggunaan sumber daya CPU hingga mendekati 100%. Seperti yang terlihat pada Tabel 6, utilitas CPU meningkat seiring penambahan *thread* yang berjalan. Akan tetapi, peningkatan utilitas CPU ini mengakibatkan penurunan pada kecepatan CPU. Hal ini kemungkinan diakibatkan oleh beberapa hal, di antaranya karena jumlah kromosom yang dieksekusi oleh setiap *thread* pada skenario 1 s.d. skenario 4 semakin sedikit sehingga CPU tidak perlu bekerja keras untuk mengeksekusi tiap *thread*. Kemungkinan lainnya, CPU dengan semakin banyak *thread* harus membagi sumber dayanya ke semua *thread* yang berjalan, sehingga hal tersebut mendorong CPU harus bekerja pada kecepatan dasar tiap *core* yang kecepatannya cenderung lebih rendah ketimbang hanya melibatkan satu atau dua *thread* saja. Hal ini dikonfirmasi oleh kecepatan CPU yang hanya berjalan pada frekuensi 1.6 GHz Ketika 90% sumber dayanya digunakan (skenario 4) di mana kecepatan tersebut sama dengan kecepatan dasar dari prosesor Intel Core i5 8250u yang digunakan di dalam penelitian ini.

Tabel 6. Tingkat utilisasi dan kecepatan CPU selama pengujian

Skenario	Utilitas CPU	Kecepatan CPU
1	23%	2.7 GHz
2	35%	2.3 GHz
3	63%	1.7 GHz
4	90%	1.6 GHz

Di samping peningkatan kinerja dari sisi waktu komputasi, sistem penjadwalan kuliah yang dikembangkan juga berhasil menyusun jadwal kuliah yang efektif tanpa ada satupun pelanggaran terhadap HC maupun SC yang telah didefinisikan sebelumnya. Pada dataset semester genap yang memiliki 878 data perkuliahan, sistem berhasil menyusun jadwal kuliah hanya dalam waktu sekitar 7 s.d. 15 menit. Sedangkan pada dataset semester ganjil yang terdiri dari 1140 perkuliahan, sistem membutuhkan waktu sekitar 40 s.d. 60 menit. Terlihat bahwa, terdapat peningkatan waktu kopmutasi yang signifikan pada penjadwalan kuliah semester ganjil yang jumlah perkuliahannya hanya selisih 262 lebih banyak dari jumlah perkuliahan di semester genap. Padahal, penambahan jumlah perkuliahan tersebut hanya sepertiga dari jumlah perkuliahan di semester genap. Hal ini menunjukkan bahwa Algoritma Genetika yang diusulkan masih memerlukan perbaikan agar mampu bekerja lebih efisien, sekalipun dengan jumlah perkuliahan yang banyak. Hasil tersebut diperoleh setelah melakukan percobaan sebanyak delapan kali untuk masing-masing dataset.

Jika dibandingkan dengan proses penyusunan jadwal perkuliahan secara manual oleh petugas Bagian Operasional (BO) universitas, sistem yang dibangun memiliki banyak kelebihan, baik dari sisi kualitas jadwal yang dihasilkan maupun dari efisiensi waktu dalam penyusunan jadwal tersebut. Tabel 7 memberikan perbandingan dalam beberapa hal antara proses penjadwalan otomatis berbasis sistem yang diusulkan pada penelitian ini dengan penjadwalan manual yang dilakukan oleh operator BO universitas.

Tabel 7. Perbandingan proses penyusunan jadwal secara otomatis menggunakan sistem dengan penjadwalan manual oleh manusia

Aspek Perbandingan	Penjadwalan Kuliah	
	Otomatis	Manual
Waktu penyusunan jadwal	7 – 60 menit	5 – 7 hari
Bentrok jadwal dosen	Tidak ada	Ada
Beban harian dosen berlebih	Tidak ada	Ada
Beban harian mahasiswa berlebih	Tidak ada	Ada
Dosen mengajar di atas pukul 12.00 di Hari Sabtu	Tidak ada	Ada

DOI: https://doi.org/10.29207/resti.v5i3.2988

4. Kesimpulan

Penelitian ini bertujuan mengembangkan model paralelisasi Algoritma Genetika Terpandu pada sistem penjadwalan kuliah universitas. Model penjadwalan kuliah yang digunakan pada penelitian ini memiliki alokasi waktu dinamis di mana durasi tiap perkuliahan bergantung pada SKS mata kuliah yang diajar. Model paralelisasi dengan skema CPU *multithreading* berhasil menurunkan waktu komputasi hingga 62% dengan penggunaan 10 *thread* untuk memproses 10 kromosom serta meningkatkan utilitas CPU. Namun, dari hasil pengujian terlihat bahwa Algoritma Genetika yang diusulkan harus bekerja ekstra keras ketika terjadi penambahan jumlah perkuliahan dari 878 perkuliahan di semester genap menjadi 1140 perkuliahan di semester ganjil. Hal ini menunjukkan bahwa AG yang diusulkan masih memerlukan perbaikan agar dapat menyusun jadwal perkuliahan dalam jumlah yang banyak (di atas 1000 perkuliahan) dengan waktu komputasi yang lebih singkat.

Ucapan Terimakasih

Terima kasih kepada Direktorat Riset dan Pengabdian Masyarakat, Deputi Bidang Penguatan Riset dan Pengembangan, Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional yang telah memberikan dukungan secara finansial sehingga penelitian ini dapat terlaksana dengan baik.

Daftar Rujukan

[1] L. Shuguang and B. Lin, "Research on Complex Curriculum Arrangement Problem Based on Novel Quantum Genetic Evolutionary Algorithm," *Proc. 31st Chinese Control Decis. Conf. CCDC 2019*, pp. 3783–3787, 2019, doi: 10.1109/CCDC.2019.8832728.

[2] Y. Yang, W. Gao, and Y. Gao, "Mathematical modeling and system design of timetabling problem based on improved GA," *ICNC-FSKD 2017 - 13th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov.*, pp. 214–220, 2018, doi: 10.1109/FSKD.2017.8393102.

[3] S. Yang and S. N. Jat, "Genetic algorithms with guided and local search strategies for university course timetabling," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 1, pp. 93–106, 2011, doi: 10.1109/TSMCC.2010.2049200.

[4] M. Lindahl, A. J. Mason, T. Stidsen, and M. Sørensen, "A strategic view of University timetabling," *Eur. J. Oper. Res.*, vol. 266, no. 1, pp. 35–45, 2018, doi: 10.1016/j.ejor.2017.09.022.

[5] Suyanto, "An Informed Genetic Algorithm for University," *Lect. Notes Comput. Sci.*, vol. vol 6114, no. 1, pp. 229–236, 2010.

[6] A. A. Gozali, J. Tirtawangsa, and T. A. Basuki, "Asynchronous island model genetic algorithm for university course timetabling," *PATAT 2014 - Proc. 10th Int. Conf. Pract. Theory Autom. Timetabling*, no. August, pp. 179–187, 2014.

[7] A. A. Gozali and S. Fujimura, "Reinforced island model genetic algorithm to solve university course timetabling," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 6, pp. 2747–2755, 2018, doi: 10.12928/TELKOMNIKA.v16i6.9691.

[8] S. Pappu, K. T. Talele, and J. Mandviwala, "Application of Parallel Genetic Algorithm for Exam Timetabling Problem," vol. 3, no. 9, pp. 1–4, 2012.

[9] K. Henryk, "Parallelisation of genetic algorithms for solving university timetabling problems," 2006.

[10] A. R. East, "Timetable Scheduling via Genetic Algorithm," 2019.

[11] 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: 10.1016/j.proeng.2012.06.033.

[12] T. Song, S. Liu, X. Tang, X. Peng, and M. Chen, "An iterated local search algorithm for the University Course Timetabling Problem," *Appl. Soft Comput. J.*, vol. 68, pp. 597–608, 2018, doi: 10.1016/j.asoc.2018.04.034.

[13] T. Mauritsius, A. N. Fajar, Harisno, and P. John, "Novel Local Searches for Finding Feasible Solutions in Educational Timetabling Problem," *Proc. 2017 5th Int. Conf. Instrumentation, Commun. Inf. Technol. Biomed. Eng. ICICI-BME 2017*, no. November, pp. 270–275, 2018, doi: 10.1109/ICICI-BME.2017.8537723.

[14] S. F. H. Irene, S. Deris, and S. Z. Mohd Hashim, "A combination of PSO and local search in university course timetabling problem," *Proc. - 2009 Int. Conf. Comput. Eng. Technol. ICCET 2009*, vol. 2, pp. 492–495, 2009, doi: 10.1109/ICCET.2009.188.

[15] D. Lohpetch and S. Jaengchuea, "A hybrid multi-objective genetic algorithm with a new local search approach for solving the post enrolment based course timetabling problem," *Adv. Intell. Syst. Comput.*, vol. 463, pp. 195–206, 2016, doi: 10.1007/978-3-319-40415-8_19.

[16] J. B. Matias, A. C. Fajardo, and R. M. Medina, "A fair course timetabling using genetic algorithm with guided search technique," *Proc. 2018 5th Int. Conf. Bus. Ind. Res. Smart Technol. Next Gener. Information, Eng. Bus. Soc. Sci. ICBIR 2018*, pp. 77–82, 2018, doi: 10.1109/ICBIR.2018.8391170.

[17] J. B. Matias, A. C. Fajardo, and R. M. Medina, "Examining Genetic Algorithm with Guided Search and Self-Adaptive Neighborhood Strategies for Curriculum-Based Course Timetable Problem," *Proc. - 2018 4th Int. Conf. Adv. Comput. Commun. Autom. ICACCA 2018*, pp. 1–6, 2018, doi: 10.1109/ICACCAF.2018.8776728.

DOI: https://doi.org/10.29207/resti.v5i3.2988