

ISSN: 9-772622-984006

semilkom.apps.cs.ipb.ac.id

SEMINAR ILMIAH ILMU KOMPUTER

Machine Learning For Agriculture 4.0

Sabtu, 12 Mei 2018 | Kampus IPB Dramaga

Prosiding



DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR

PROSIDING SEMINAR ILMIAH ILMU KOMPUTER 2018

Machine learning for Agriculture 4.0

Hak cipta dilindungi undang-undang

Copyright @2018

ISSN: 9-772622-984006

Dipublikasikan oleh:

Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Pertanian Bogor
Bogor – Indonesia

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT, karena atas karunia-Nya Prosiding Seminar Ilmiah Ilmu Komputer 2018 ini dapat diterbitkan. Seminar dengan tema “Machine Learning for Agriculture 4.0” dilaksanakan pada tanggal 12 Mei 2018 di Auditorium Common Class Room (CCR), Kampus IPB Dramaga - Bogor, dengan penyelenggara Departemen Ilmu Komputer, FMIPA IPB.

Seminar ini diselenggarakan sebagai media bertukar gagasan dan sosialisasi hasil penelitian para akademisi, peneliti, birokrat, dan mahasiswa. Seminar ini juga sebagai ajang untuk memunculkan pendekatan-pendekatan baru dalam rangka melakukan optimalisasi pemanfaatan metode komputasi dan teknologi informasi untuk mendukung pertanian. Diharapkan dari kegiatan seminar ini akan muncul peluang-peluang dalam bentuk sinergisitas penelitian yang melibatkan berbagai institusi untuk ‘membangkitkan’ teknologi informasi ke dalam domain-domain pertanian.

Buku prosiding ini merupakan kumpulan makalah yang dipresentasikan pada Seminar Ilmiah Ilmu Komputer tahun 2018. Prosiding ini dibagi menjadi empat bagian makalah, yaitu Bidang Bioinformatika, Bidang Data Science, Bidang Network Computing, Bidang Sistem Informasi, dan Bidang Aplikasi Teknologi untuk Pertanian.

Semoga penerbitan prosiding ini dapat digunakan sebagai salah satu rujukan dalam pengembangan penelitian di masa yang akan datang, serta dijadikan bahan acuan dalam bidang ilmu komputer. Akhir kata kepada semua pihak yang telah membantu, kami ucapkan terima kasih.

Bogor, Agustus 2018

Ketua Pelaksana

Dr Imas Sukaesih Sitanggang, S.Si M.Kom

**EDITOR DAN REVIEWER
PROSIDING SEMINAR ILMIAH ILMU KOMPUTER
2018**

Direktor Editor

Wulandari, S.Komp., M.AgrSc

Husnul Khotimah, S.Komp., M.Kom

Dean Apriana Ramadhan, S.Komp., M.Komp

Reviewer

Dr Imas Sukaesih Sitanggang, S.Si M.Kom

Lailan Sahrina Hasibuan, S.Kom., M.Kom

Muhammad Asyhar Agmaloro, S.Si, M.Kom

Dr. Medria Kusuma Dewi Hardhienata, S.Komp

DAFTAR ISI

KATA PENGANTAR	iv
EDITOR DAN REVIEWER.....	v
BIOINFORMATIKA.....	1
Pengukuran Kemiripan Protein pada Ijah Webserver dengan Algoritme Smith Waterman Berbasis Komputasi Paralel Cuda GPU	2
Analisis Kinerja Basis Data NoSQL pada Basis Data Ijah	11
Pengujian <i>Usability</i> pada Ijah-Webserver dengan Menggunakan Metode Cognitive Walkthrough.....	21
DATA SCIENCE.....	29
Algoritme A* untuk Pencarian Jalur Terpendek dari Serdang Baru, Kemayoran Menuju Mall Taman Anggrek	30
Komparasi Metode <i>Machine Learning</i> Pada Prediksi Kelulusan Mahasiswa.....	37
Perbandingan Metode C-PCA, PP-PCA dan ROB-PCA dalam Menganalisis Data Pencilan	46
Model Regresi Logistik Kekar Pada Status <i>Stunting</i> : Perbandingan Beberapa Metode	54
Penjadwalan Ujian Pendadaran Proyek Tugas Akhir Menggunakan Algoritme Genetika Dengan <i>Repetitive Random Approach</i>	61
Implementasi Algoritme <i>Clustering</i> Dengan <i>Singular Vector Decomposition</i> Untuk Menunjang Keputusan Dalam Meningkatkan Produktivitas Tanaman Jagung	71
Klasifikasi Kemunculan Titik Panas pada Lahan Gambut di Sumatera dan Kalimantan Menggunakan Algoritme <i>K-Nearest Neighbor</i>	79
Modul <i>Incremental Sequential Pattern Mining</i> untuk Data Titik Panas Pulau Sumatra.....	88
<i>Post Pruning</i> Pohon Keputusan untuk Klasifikasi Kemunculan Titik Panas.....	97
NETWORK COMPUTING.....	105
Monitoring Keamanan Jaringan Internet dan Aplikasi dengan SIMONTIK	106
Otomasi Instalasi Dan Konfigurasi Komputer Pada Lab Ilmu Komputer IPB Dengan Saltstack.....	110
Pengembangan Fitur Diskusi Waktu Nyata Pada Aplikasi KMS Kedelai Berbasis Protokol WebRTC.....	118
SISTEM INFORMASI	125
Penerapan Metode <i>Simple Additive Weighting</i> (SAW) pada Sistem Pendukung Keputusan Penerima Bantuan Simpatik Guru	126
SIGEPOK : Sistem Informasi Geografis Rute Angkutan Umum Kota Depok Berbasis Web menggunakan Metode RAD	138
Sistem Informasi Manajemen Sarana Dan Prasarana Pusat Studi Biofarmaka Tropika IPB (Trop Brc).....	150
Pengembangan <i>Back-end</i> Bagian <i>Provider</i> pada <i>Marketplace</i> TRAVINESIA.COM dengan REST API.....	160
Pengembangan <i>Back-end</i> Sistem Informasi <i>Marketplace</i> Pariwisata TRAVINESIA.COM dengan REST API dari Sisi <i>User</i>	170

Pengembangan Aplikasi Repositori Studi Kasus Dinas Kependudukan dan Catatan Sipil Kota Bogor untuk Mendukung Smart City	180
Pengembangan Sistem Informasi <i>Billing</i> Elektronik untuk Penyebaran <i>Invoice</i> dan Kartu Piutang Menggunakan Metode <i>Prototyping</i> (Studi Kasus: PT. XYZ)	188
Pengembangan <i>Front-end</i> Bagian <i>User</i> Sistem Informasi <i>Marketplace</i> TRAVINESIA.COM Berbasis Web	197
Sistem Informasi Manajemen Peneliti untuk Manajemen Puncak di Pusat Studi Biofarmaka Tropika (Trop BRC).....	207
Pengembangan Sistem Informasi Manajemen Pelaporan Keuangan di Pusat Studi Biofarmaka Tropika	215
Sistem Informasi Sumber Daya Manusia pada Pusat Studi Biofarmaka Tropika IPB	223
Pengembangan Sistem Informasi Penelitian Pusat Studi Biofarmaka Tropika IPB untuk <i>Low Level Management</i> dan Umum	232
Aplikasi Penggabungan Poligon Menggunakan <i>Web Framework</i> Shiny pada Bahasa Pemrograman R.....	240
Pembuatan Aplikasi <i>Mobile</i> Siptiket PT Versa Technology.....	248
Sistem Informasi Verifikasi Soal Ujian di Departemen Ilmu Komputer IPB.....	258
Pengembangan Fitur Cuti pada Sistem Informasi Karyawan di PT XYZ	267
Evaluasi dan Pengembangan <i>Digital Literacy</i> Staf Pemerintah berdasarkan <i>BC's Digital Literacy Framework</i>	276
Sistem Manajemen Pengelolaan Dokumen dan Rekaman ISO 9001:2008 Berbasis <i>Web</i> Untuk Menunjang Kegiatan Audit.....	286
<i>Local E-Government</i> : Pengembangan Sistem Pelayanan Publik Kecamatan Menggunakan Konsep Arsitektur Informasi.....	296
APLIKASI TEKNOLOGI INFORMASI UNTUK PERTANIAN	305
Analisis Perancangan Sistem Identifikasi Jamur Menggunakan Jaringan Perceptron.....	306
Pemantauan Suhu, Kelembaban dan CO ₂ /NH ₃ Pada Kandang Ayam Petelur Sistem Tertutup menggunakan Internet of Things	316
Visualisasi Hasil <i>Clustering</i> Data Berbasis Media Sosial (Studi Kasus: Data Pertanian) .	324
OLAP untuk Tanaman Holtikultura Menggunakan Palo.....	333
Pengembangan <i>Back End Knowledge Management System</i> pada Aplikasi Digital Tani ..	342
Pengembangan Aplikasi KMS Kedelai Berbasis <i>Mobile</i> Menggunakan Pendekatan UX	351
Pengujian Usability Pada Knowledge Management System Kedelai	360
Prototipe Sistem Informasi Geografi Pemanfaatan Hutan Di Indonesia	367

DATA SCIENCE

Penjadwalan Ujian Pendadaran Proyek Tugas Akhir Menggunakan Algoritme Genetika Dengan *Repetitive Random Approach*

Final Project Examination Scheduling Using Genetic Algorithm with Repetitive Random Approach

Adityo Permana Wibowo*, Donny Avianto

Program Studi Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Daerah Istimewa Yogyakarta,
55285, Indonesia,

Email: *adityopw@staff.uty.ac.id, donny@staff.uty.ac.id

Abstrak

Ujian pendadaran proyek tugas akhir adalah fase yang biasanya dilalui seorang mahasiswa sebelum mendapatkan gelar akademik untuk jenjang Strata-1 (S-1). Pada pendadaran tersebut mahasiswa akan berusaha untuk mempertahankan tugas akhir buatannya di hadapan tim penguji yang terdiri dari dosen penguji dan dosen pembimbing. Permasalahan yang timbul adalah penjadwalan ujian pendadaran yang harus menyesuaikan waktu beberapa dosen penguji, ketersediaan ruangan dan sesi untuk ujian pendadaran. Algoritme Genetika (GA) merupakan salah satu metode optimasi yang biasa digunakan untuk melakukan penjadwalan, termasuk penjadwalan ujian pendadaran. Namun GA secara umum cenderung kesulitan menemukan solusi untuk permasalahan kompleks seperti penjadwalan ujian pendadaran dengan cepat. Oleh karena itu, pada penelitian ini diusulkan suatu pendekatan yang memodifikasi GA secara umum. Modifikasi dilakukan pada tahap mutasi gen dengan menambahkan konsep *repetitive random (RR)*. Adapun representasi gen yang digunakan pada penelitian ini bisa menampung beberapa info seperti: dosen penguji, mahasiswa, ruangan, sesi, dan hari. Setelah melakukan uji coba, diketahui sistem dengan pendekatan *RR* membutuhkan rata-rata waktu eksekusi 2.30 detik untuk menemukan 2 solusi dengan nilai $n=7$. Dari penelitian ini, diketahui juga bahwa frekuensi mutasi mempengaruhi jumlah generasi yang dihasilkan untuk mencari solusi. Dengan kata lain, penggunaan konsep *RR* dan seringnya proses mutasi akan lebih cepat menghasilkan solusi dibandingkan GA pada umumnya.

Kata Kunci: Algoritme Genetika, Penjadwalan, *Repetitive Random*, Ujian Pendadaran

Abstract

Final project exam is a stage student usually passed through to obtain academic degree in undergraduate degree (S1). On that exam, student will make attempt to defense his final project in front of boards of examiners, included examiner lecturer and advisor lecturer. The problem occurred then, is that how the final exam schedule which should adjust several lecturer's teaching schedule, room availability and final project exam sessions. Genetics Algorithm (GA) is one of the optimization methods, commonly used for scheduling, one of those is final exam scheduling. However, GA generally tends to be difficult in finding solution for more advanced problem such as quick scheduling. This research proposes an approach which modify GA for scheduling problems. Modification done through the mutation process by accumulating repetitive random (RR) concept. Gen representation which used in this approach could accommodate information such as examiner lecturer, students, rooms, sessions and days. By doing an experiment, it is obtained that system using RR approach needs average execution time for 2.30 seconds to find 2 solutions on $n=7$. It is acknowledged from this research that mutation frequency can influence the amount of generation resulted to find solution. So that the use of RR concept and the frequent mutation process can obtain solution faster than the common GA.

* Penulis korespondensi

PENDAHULUAN

Ujian pendadaran proyek tugas akhir adalah fase yang biasanya dilalui seorang mahasiswa sebelum mendapatkan gelar akademik untuk jenjang Strata-1 (S-1). Pada pendadaran tersebut mahasiswa akan berusaha untuk mempertahankan tugas akhir buatannya dihadapan tim penguji yang terdiri dari dosen penguji dan dosen pembimbing. Masalah administratif yang kemudian muncul saat akan menyelenggarakan suatu ujian pendadaran adalah membuat jadwal. Membuat jadwal untuk penyelenggaraan pendadaran proyek tugas akhir bisa menjadi kompleks karena beberapa faktor. Faktor pertama, seorang pembuat jadwal harus memperhatikan jadwal kegiatan dosen yang akan bergabung sebagai tim penguji. Mengingat padatnya jadwal kegiatan dosen, seorang pembuat jadwal harus memperhatikan kegiatan dosen di dalam dan di luar kampus. Tidak jarang juga seorang pembuat jadwal harus melakukan komunikasi secara intens melalui telepon atau pesan singkat untuk mengetahui jadwal kegiatan seorang dosen. Kompleksitas untuk membuat jadwal juga akan bertambah seiring dengan bertambahnya jumlah dosen yang akan bergabung sebagai tim penguji. Faktor kedua yang menambah kompleksitas pembuatan jadwal adalah jumlah ruang dan sesi yang tersedia untuk menyelenggarakan ujian pendadaran proyek tugas akhir. Seringkali dijumpai pada suatu institusi akademik, suatu ruangan digunakan untuk menyelenggarakan ujian pendadaran dari program studi atau fakultas yang berbeda-beda. Hal ini mewajibkan seorang pembuat jadwal untuk berkoordinasi juga dengan pihak-pihak lain yang memiliki hak menggunakan ruangan tersebut. Jika seorang pembuat jadwal tidak memperhatikan kedua faktor ini, maka bisa jadi jadwal pendadaran yang sudah dibuat harus dibatalkan dan dibuat ulang karena adanya terbentur dengan jadwal lainnya. Membuat ulang suatu jadwal pendadaran juga memerlukan usaha yang tidak sedikit karena harus memastikan jadwal yang dibuat tidak berbenturan dengan jadwal lainnya. Hal ini juga memberikan dampak pada lamanya waktu tunggu mahasiswa untuk mendapatkan jadwal ujian pendadaran.

Salah satu metode yang sering digunakan untuk menyelesaikan permasalahan kompleks seperti di atas adalah *Genetic Algorithm*. *Genetic Algorithm* (GA) adalah metode optimasi yang meniru perilaku evolusi biologis untuk memecahkan suatu masalah tertentu secara terprogram (Obaid *et al.*, 2012). Pada awalnya GA hanya digunakan sebagai algoritme untuk mencari nilai parameter yang optimal. Namun seiring dengan perkembangan zaman, saat ini GA biasa diaplikasikan pada berbagai konteks permasalahan seperti: pembelajaran mesin, peramalan, pemrograman otomatis, dan sebagainya. Penerapan GA untuk mencari solusi dari suatu permasalahan memiliki beberapa kelebihan antara lain (Tabassum, 2014) :

1. GA mampu menghasilkan solusi dari sembarang permasalahan optimasi dengan pendekatan kromosom.
2. GA mampu menangani permasalahan dipengaruhi oleh banyak faktro dan memiliki banyak solusi.
3. GA lebih lebih sederhana dan lebih mudah digunakan daripada algoritme optimasi lainnya.
4. GA memiliki tingkat fleksibilitas yang tinggi karena dapat diterapkan pada platform yang berbeda-beda dengan mudah.

Kemampuan GA untuk menemukan solusi yang optimal di berbagai bidang sudah banyak dibuktikan melalui berbagai penelitian, seperti: penelitian di bidang distribusi logistik untuk menyelesaikan permasalahan *Travelling Salesman Problem* (Borna and Hashemi, 2014) dan (Taiwo, Mayowa and Ruka, 2013), di bidang biologi untuk optimalisasi parameter dalam deteksi inti sel (Szénási and Vámosy, 2013), di bidang rekayasa perangkat lunak untuk meakukan pengujian perangkat lunak (Sharma, Patani and Aggarwal, 2016), dan di bidang ekonomi untuk menentukan parameter pada fungsi produksi Cobb-Douglas (Mateescu, 2010).

Selain digunakan untuk mencari solusi dari permasalahan optimalisasi, GA juga sangat handal untuk menangani masalah penjadwalan. Beberapa jadwal yang berhasil dibuat

menggunakan GA antara lain adalah: jadwal penggunaan sumber daya pada perusahaan *well-service* (Brezulianu, Fira and Fira, 2012), jadwal transit untuk kendaraan umum (Johar, Jain and Garg, 2016), jadwal jaga perawat di rumah sakit (Kim *et al.*, 2014) dan jadwal penggunaan mesin di pabrik (Li and Chen, 2010) dan (Nazif, 2012). Di Indonesia sendiri, metode GA sering diaplikasikan untuk membuat jadwal di suatu institusi pendidikan seperti sekolah atau universitas.

Penelitian (Saryanti and Wijanegara, 2017) yang menggunakan GA untuk pembuatan jadwal mata pelajaran di suatu Sekolah Menengah Kejuruan (SMK). Beberapa *constraint* yang digunakan pada penelitian ini antara lain: jumlah maksimal mengajar dalam satu hari, daftar mata pelajaran dan guru pengampunya, serta definisi satu jam pelajaran. Namun, pada penelitian ini tidak ada penjelasan seputar representasi kromosom yang digunakan untuk menghasilkan jadwal. Pembahasan secara mendetil hanya dilakukan pada bagian penentuan beban mengajar guru dan bukan pada proses evolusi untuk menghasilkan jadwal mengajar. Meskipun hasil eksperimen pada penelitian ini menunjukkan sistem yang dibuat mampu menghasilkan jadwal mengajar yang cukup efektif dan efisien, penelitian ini tidak menunjukkan luaran asli dari sistem dalam bentuk jadwal mengajar.

Penelitian serupa juga pernah dilakukan oleh (Puspaningrum, Djunaidy and Vinarti, 2013). Pada penelitian ini, GA digunakan untuk menghasilkan solusi berupa jadwal mata kuliah di suatu institut. Penelitian ini memberikan batasan bahwa jadwal yang dihasilkan adalah jadwal mata kuliah untuk satu jurusan saja yaitu Sistem Informasi. Adapun beberapa *constraint* yang digunakan pada saat pembuatan jadwal menggunakan GA antara lain adalah: jumlah dosen, jumlah mahasiswa, jumlah mata kuliah, serta ketersediaan waktu dan ruang kelas. Meskipun memiliki *constraint* yang cukup lengkap untuk menghasilkan jadwal yang baik, penelitian ini juga tidak membahas secara mendalam terkait representasi gen yang digunakan selama proses evolusi terjadi. Selain itu, penelitian ini juga tidak membahas secara detil fungsi *fitness* yang digunakan untuk mengukur kesesuaian kromosom yang dihasilkan dari proses evolusi. Dari hasil pengujian yang dilakukan pada penelitian ini, diketahui bahwa untuk menghasilkan jadwal yang baik pada satu jurusan dibutuhkan waktu ± 3 menit.

Penelitian yang menerapkan GA untuk membuat jadwal di perguruan tinggi juga pernah dilakukan oleh (Suhartono, 2015). Namun, sedikit berbeda dengan penelitian sebelumnya, pada penelitian ini, jadwal yang dihasilkan oleh GA adalah jadwal penggunaan laboratorium komputer. Pada penelitian ini, *constraint* yang digunakan antara lain adalah: hari aktif perkuliahan, bobot SKS mata kuliah yang akan diselenggarakan, dan aturan bahwa seorang dosen dan mahasiswa hanya bisa berada di satu kelas pada satu waktu. Penelitian ini menggunakan kromosom yang terdiri dari 60 gen dengan asumsi ruangan laboratorium yang akan dijadwalkan hanya satu ruangan. Sedangkan pada fungsi *fitness* yang digunakan berlaku sifat seperti poin pinalti, dimana setiap aturan pada *constraint* dilanggar maka akan diberikan nilai pinalti. Berdasarkan pengujian yang dilakukan pada penelitian ini, terlihat bahwa metode GA yang digunakan mampu menghasilkan jadwal sesuai dengan *constraint* yang telah ditetapkan. Namun pada penelitian ini tidak dibahas berapa lama waktu yang digunakan untuk menghasilkan solusi dan berapa nilai *fitness* akhir yang dihasilkan oleh solusi.

Berdasarkan beberapa penelitian yang telah dibahas di atas, belum terlihat peneliti yang menggunakan metode GA untuk menghasilkan jadwal ujian pendadaran Tugas Akhir mahasiswa. Oleh sebab itu, tujuan yang akan dicapai pada penelitian ini adalah menghasilkan generasi dan solusi terbaik dari Algoritme Genetika yang nantinya akan digunakan untuk penjadwalan ujian pendadaran Proyek Tugas Akhir. Bagian selanjutnya dalam artikel ini akan membahas tentang metode yang digunakan pada penelitian kali ini, hasil dan pembahasan, serta yang terakhir adalah kesimpulan penelitian.

METODE

Bagian ini menjelaskan tentang metode yang digunakan untuk menyelesaikan masalah penjadwalan tugas akhir. Pada penelitian ini metode yang dibahas antara lain : *constraint* atau kendala yang digunakan, representasi gen yang digunakan, dan usulan sistem GA yang menerapkan konsep *repetitive random (RR)*.

Kendala-kendala (*Constraints*) yang digunakan dalam proses Algoritme Genetika

Menurut (Burke and Petrovic, 2002) terdapat dua *constraint*/batasan dalam penyusunan penjadwalan yaitu *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi). *Hard constraint* adalah batasan yang harus terpenuhi sedangkan *soft constraint* adalah batasan yang tidak selalu terpenuhi dalam proses pembentukan jadwal. Meskipun sifatnya tidak harus terpenuhi, solusi yang dihasilkan sebaiknya memenuhi ketentuan *soft constraint*.

Oleh karena itu perlu ditetapkan suatu batasan dalam penyusunan jadwal yang bersifat harus dipenuhi (*hard constraints*) dan tidak harus dipenuhi (*soft constraints*), tetapi tetap menjadi acuan dalam proses pembuatan jadwal (Mawaddah and Mahmudy, 2006).

Penelitian ini menggunakan *constraint* yang dimaksudkan agar model bisa menghasilkan sistem yang sesuai dengan peraturan yang sudah ditetapkan di Program Studi Teknik Informatika Universitas Teknologi Yogyakarta sebelumnya. *Constraint* ini juga digunakan untuk menentukan nilai *fitness* untuk perhitungan algoritme genetika. Adapun poin-poin yang penulis tetapkan sebagai *constraint* pada penelitian ini adalah sebagai berikut :

1. Total mahasiswa yang dapat dijadwalkan dalam 1 kali menjalankan GA pada sistem ini adalah 20 mahasiswa.
2. Tidak boleh ada mahasiswa yang dijadwalkan lebih dari satu kali di dalam 1 individu kromosom dan setiap siswa harus mendapatkan 1 slot jadwal pada setiap individu kromosom.
3. Jumlah dosen yang harus ada di ruang ujian pendadaran adalah 3 orang.
4. Jumlah ruangan yang bisa digunakan secara bersamaan untuk ujian pendadaran dalam satu waktu adalah dua ruangan.
5. Tidak boleh ada dosen yang dijadwalkan di 2 ruang kelas pada waktu yang bersamaan.

Dari kelima *constraint* tersebut dibuat dalam bentuk komponen Algoritme Genetika. Terdapat 10 orang dosen yang sudah berhak untuk menjadi dosen penguji ujian pendadaran. Teknis pelaksanaan ujian pendadaran adalah dalam 5 hari kerja (senin s/d jumat) dan hanya menggunakan 2 ruangan. Dalam satu hari ujian pendadaran terdapat 4 sesi ujian pendadaran, sehingga dalam 1 ruangan digunakan untuk 2 sesi ujian pendadaran.

Representasi Gen

Berdasarkan teknis pelaksanaan ujian pendadaran yang sudah dijelaskan, maka komponen Algoritme Genetika yang digunakan pada penelitian ini terdiri dari lima Gen yang membentuk menjadi sebuah subkromosom. Satu kromosom terdiri dari empat subkromosom. Satu individu mempunyai lima kromosom, dan satu populasi terdiri dari sepuluh individu. Lima gen tersebut digunakan untuk pengelompokkan kode dosen (3 gen), kode mahasiswa (1 gen), dan kode ruangan (1 gen). seperti terlihat pada Gambar 1.



Ket :

D1 : Dosen Ketua Penguji

D2 : Dosen Penguji 1

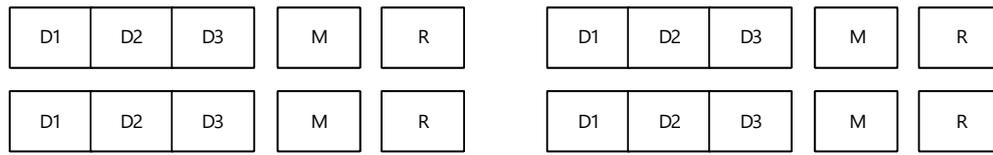
D3 : Dosen Penguji 2

M : Mahasiswa

R : Ruangan

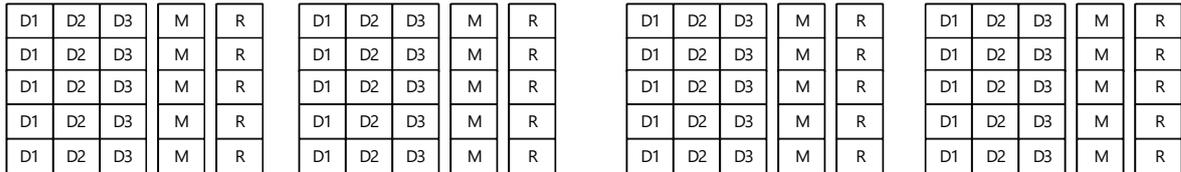
Gambar 1. Gen yang membentuk 1 subkromosom

Subkromosom yang terbentuk, dikelompokkan untuk menjadi sebuah kromosom. Dalam penelitian ini satu kromosom terdiri dari empat subkromosom, seperti terlihat pada Gambar 2.



Gambar 2. Subkromosom yang membentuk 1 kromosom

Setelah pengelompokkan kromosom, dikelompokkan menjadi sebuah individu. Dalam penelitian ini satu individu terdiri dari lima kromosom, seperti terlihat pada Gambar 3.

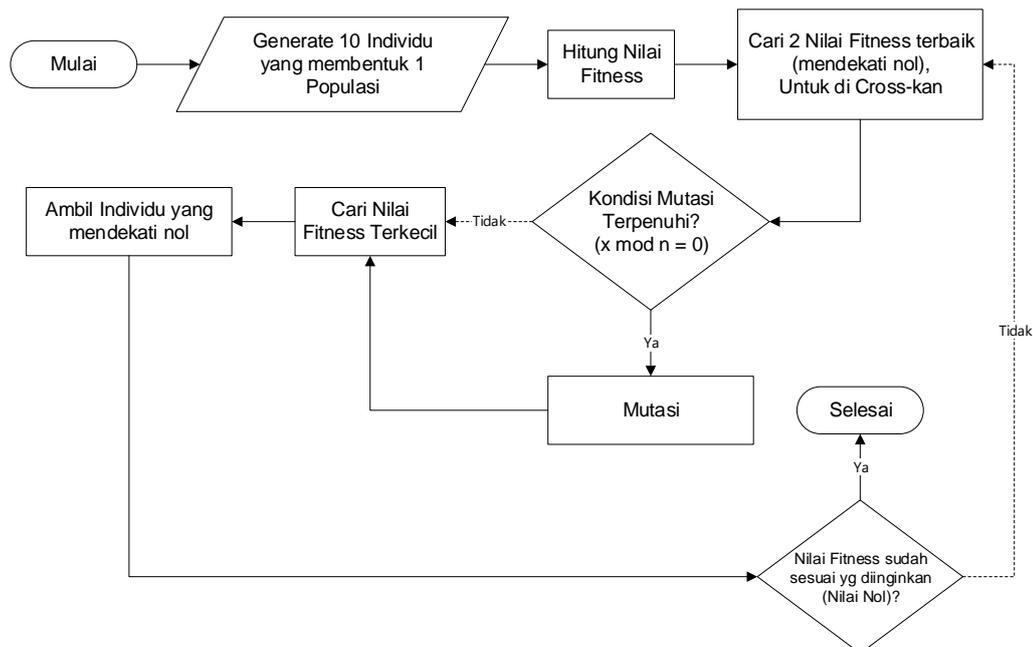


Gambar 3. Kromosom yang membentuk 1 Individu

Sistem yang diusulkan

Secara umum Algoritme Genetika diawali dengan men-*generate* 10 individu (1 populasi). Berikutnya, masing-masing individu di dalam populasi akan dihitung nilai *fitness*-nya untuk mencari 2 individu dengan nilai *fitness* yang terbaik (nilai mendekati nol). Untuk dikawinsilangkan (*cross over*). Proses *cross over* akan menghasilkan 2 individu baru (*offspring*). *Offspring* nilai *fitness* terbaik akan menggantikan individu dengan nilai *fitness* yang paling buruk (nilai *fitness* negatif) di dalam populasi.

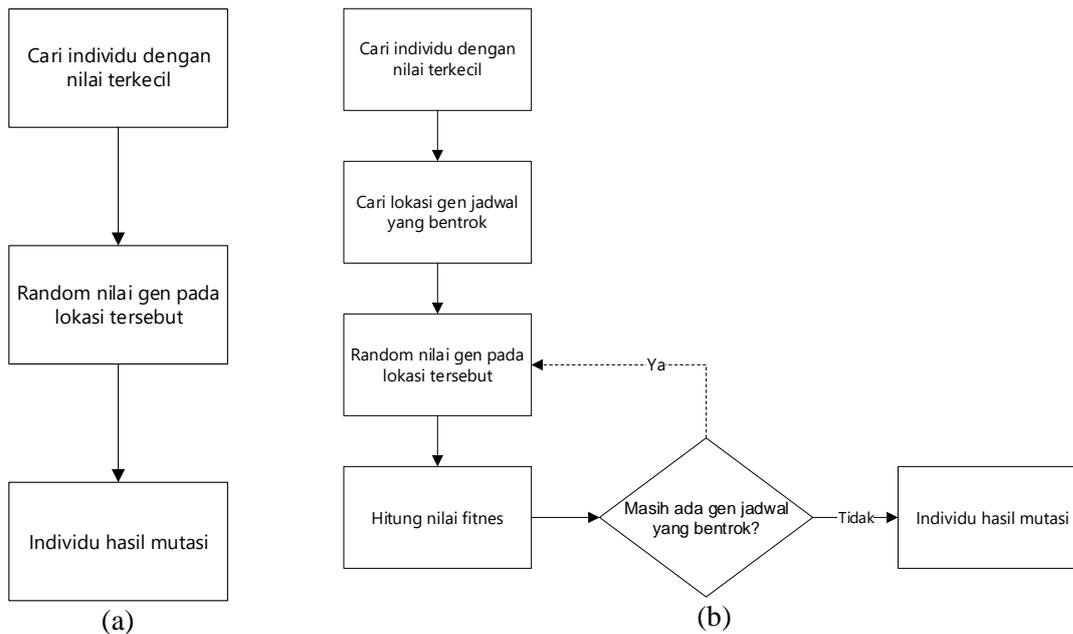
Selanjutnya sistem akan mengevaluasi nilai $x \bmod n$, dimana x adalah bilangan bulat hasil random dan n adalah bilangan bulat yang menentukan frekuensi mutasi sistem, maka mutasi akan dilakukan. Terakhir, sistem akan menghitung lagi nilai *fitness* setiap individu dan jika ada individu dengan nilai *fitness* = 0 maka sistem akan berhenti, namun jika tidak, maka sistem mengulang proses. Alur Algoritme Genetika secara umum terlihat pada Gambar 4.



Gambar 4. Flowchart pencarian komposisi terbaik pada Algoritme Genetika

Sistem algoritme genetika secara umum pada Gambar 4, diindikasikan kesulitan menemukan solusi untuk permasalahan kompleks, seperti permasalahan jadwal ujian pendadaran proyek tugas akhir.

Untuk itu pada penelitian ini, diusulkan suatu pendekatan baru yang memodifikasi GA untuk yang dapat menyelesaikan permasalahan kompleks. Modifikasi dilakukan pada proses mutasi dengan menambahkan konsep *repetitive random (RR)*. Pada proses mutasi di sistem GA yang umum, setiap gen dari masing-masing individu memiliki peluang yang sama untuk dimutasi (diubah nilai gennya). Hal ini menyebabkan hasil mutasi bisa menghasilkan individu dengan nilai fitness yang lebih baik atau lebih buruk. Sedangkan proses mutasi dengan konsep *RR*, hanya melakukan mutasi gen pada lokasi *gen* jadwal yang bentrok. Perbedaan proses mutasi antara GA secara umum dengan GA menggunakan *RR*, dapat dilihat pada Gambar 5.



Gambar 5. Perbedaan proses mutasi (a) GA secara umum (b) GA menggunakan RR

HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas hasil penelitian sesuai dengan metode yang sudah dijabarkan pada bagian sebelumnya. Mula-mula sistem akan *generate* 1 populasi yang terdiri dari 10 individu. Adapun contoh hasil representasi gen pada individu yang di-*generate* oleh sistem seperti terlihat pada Gambar 6.

```

Individu 1
1- 2- 5- 9- 1| 5-10- 8-17- 1|10- 8- 9- 3- 2| 5- 9-10- 5- 2|
7- 2- 6-18- 1| 8- 7- 5-16- 1| 5-10- 8-15- 2|10- 7- 5- 6- 2|
9- 2- 4-19- 1| 5- 2- 8-10- 1| 3-10- 6- 7- 2| 4- 7- 3-20- 2|
1- 2- 7-12- 1| 1- 8- 2- 1- 1|10- 9- 5-11- 2| 9-10- 3- 8- 2|
6- 1- 4-14- 1| 6- 1- 4- 2- 1| 2- 8- 7-13- 2| 9- 5- 3- 4- 2|
Fitness: -2
  
```

Gambar 6. Representasi Gen pada Individu

Berdasarkan Gambar 6, terlihat komposisi jadwal yang dituliskan dalam bentuk angka. Terdapat 5 baris yang menunjukkan hari pelaksanaan ujian pendadaran, yaitu hari Senin s/d Jumat. Terdapat 4 kolom yang menunjukkan sesi dan ruangan ujian pendadaran. Garis (-) di depan angka bukan sebagai tanda nilai angka (minus) melainkan digunakan sebagai pemisah angka. Nilai fitness yang dihasilkan (-2) berdasarkan perhitungan *constraint* yang artinya terdapat 2 orang dosen yang terjadwal lebih dari satu sesi/ruang (bentrok). Kode dosen yang

bentrok ditunjukkan dengan kode 10 yang lebih dari 1 hari yaitu hari senin sesi 2 ruang 1 dan 2. Kode dosen 7 dan 5 yang bentrok pada hari selasa sesi 2 ruang 1 dan 2.

Untuk mengevaluasi kinerja sistem, dilakukan 2 jenis pengujian. Pengujian pertama bertujuan membandingkan performa antara sistem GA secara umum dengan sistem GA yang sudah menerapkan *repetitive random (RR)*. Sedangkan pengujian yang kedua bertujuan untuk melihat pengaruh nilai n sebagai variabel yang menentukan mutasi dengan jumlah solusi yang ditemukan pada sistem yang menggunakan pendekatan *repetitive random (RR)*.

Pengujian pertama dimulai dengan menjalankan sistem yang belum menerapkan *repetitive random*. Pengujian ini dijalankan dengan batasan generasi mulai dari 1000 generasi hingga 10.000 generasi. Nilai n yang digunakan sebagai syarat mutasi pada pengujian ini adalah $n=7$. Dari hasil pengujian, diketahui bahwa proses pencarian solusi yang berjalan sampai dengan generasi ke-10.000 tidak menghasilkan kromosom solusi (nilai fitness bernilai nol). Hasil lengkap dari pengujian ini terlihat pada Tabel 1.

Tabel 1. Percobaan Pencarian Solusi Menggunakan Non-Repetitive Random

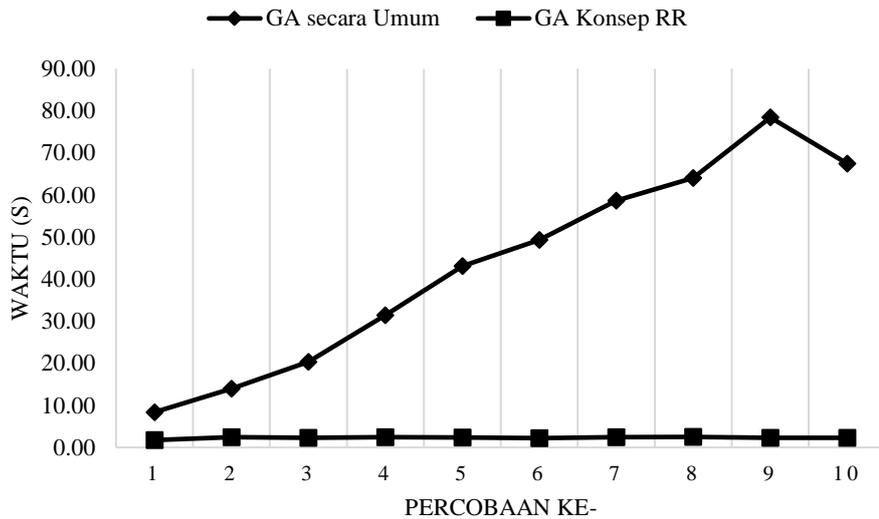
Percobaan ke-	Batas Maksimum Generasi	Iterasi yang dijalankan	Jumlah solusi yang ditemukan	Waktu Eksekusi (detik)
1	1000	1000	0	8.38
2	2000	2000	0	13.94
3	3000	3000	0	20.37
4	4000	4000	0	31.45
5	5000	5000	0	43.12
6	6000	6000	0	49.30
7	7000	7000	0	58.68
8	8000	8000	0	64.06
9	9000	9000	0	78.43
10	10000	10000	0	67.46

Selanjutnya, 10 percobaan juga dilakukan pada sistem yang menggunakan *RR*. Hasil pengujian kali ini menunjukkan, sistem *RR* jauh lebih unggul dibandingkan sistem GA secara umum dalam hal menemukan solusi. Hasil lengkap pengujian menggunakan sistem *RR* ini dapat dilihat pada Tabel 2.

Tabel 2. Percobaan Pencarian Solusi Menggunakan *Repetitive Random n=7*

Percobaan ke-	Batas Maksimum Generasi	Iterasi yang dijalankan	Jumlah solusi yang ditemukan	Waktu Eksekusi (detik)
1	1000	35	1	1,75
2	2000	82	1	2,43
3	3000	69	2	2,30
4	4000	73	1	2,46
5	5000	80	1	2,36
6	6000	56	1	2,25
7	7000	68	1	2,44
8	8000	74	1	2,50
9	9000	67	2	2,28
10	10000	50	1	2,26

Berdasarkan Tabel 1 dan Tabel 2 di atas, dari 10 percobaan yang dilakukan pada masing-masing sistem, sangat jelas terlihat bahwa sistem dengan *RR* memiliki performa yang lebih baik dibandingkan sistem GA pada umumnya. Hal ini terbukti dari rata-rata waktu yang dibutuhkan oleh *RR* dalam menemukan solusi yaitu 2.30 detik. Sedangkan pada sistem GA pada umumnya masih belum menemukan solusi meskipun sudah berjalan selama 1 menit. Selain itu, sistem dengan konsep *RR* selalu menyediakan minimal 1 solusi pada generasi terakhir. Adapun grafik perbandingan waktu proses untuk 10 percobaan dapat dilihat pada Gambar 7.



Gambar 7. Grafik Perbandingan Waktu untuk Menemukan Solusi

Pengujian kedua akan berfokus pada sistem *RR*. Pada pengujian ini akan dicari pengaruh frekuensi mutasi terhadap kecepatan sistem menghasilkan solusi. Untuk itu nilai variabel n akan divariasikan dan dieksekusi sebanyak 10 kali dengan maksimal 1000 iterasi. Nilai variasi n berturut-turut adalah 7, 50, 75, 100. Nilai n tersebut diambil dengan asumsi semakin besar nilainya maka semakin jarang mutasi yang dilakukan oleh sistem. Semakin jarang mutasi banyak generasi yang harus di-*generate* oleh sistem untuk mendapatkan solusi. Adapun hasil lengkapnya dapat dilihat pada Tabel 3.

Tabel 3. Pengaruh Frekuensi Mutasi Terhadap Jumlah Generasi untuk Mendapatkan Solusi Variasi nilai n

Percobaan ke-	Variasi nilai n							
	n=7		n=50		n=75		n=100	
	Jumlah solusi ditemukan	Iterasi yang dijalankan						
1	1	69	1	814	1	138	1	273
2	2	13	4	321	2	371	4	434
3	1	35	1	270	1	198	1	661
4	4	39	2	447	2	627	3	793
5	1	57	1	630	4	658	8	172
6	2	71	1	561	1	560	2	905
7	4	57	4	350	2	560	6	918
8	3	41	2	462	2	300	2	879
9	1	44	2	23	4	934	6	683
10	1	32	2	236	2	488	4	450

Berdasarkan Tabel 3, ditampilkan perbedaan hasil rata-rata solusi dan generasi yang menghasilkan solusi berdasarkan nilai n . Sistem yang menggunakan nilai $n=7$, secara rata-rata, menemukan solusi pada generasi ke-45.8 dan jumlah solusi yang ditemukan adalah 2 solusi. Untuk sistem yang menggunakan nilai $n=50$, secara rata-rata, menemukan solusi pada generasi ke-411.4 dan jumlah solusi yang ditemukan adalah 2 solusi. Untuk sistem yang menggunakan nilai $n=75$, secara rata-rata, menemukan solusi pada generasi ke-483.4 dan jumlah solusi yang ditemukan adalah 2.1 solusi. Sedangkan untuk sistem yang menggunakan nilai $n=100$, secara rata-rata, menemukan solusi pada generasi ke-616.8 dan jumlah solusi yang ditemukan adalah 3.7 solusi. Hasil pengujian kedua ini menunjukkan bahwa asumsi yang digunakan di awal bahwa semakin sering sistem *RR* melakukan mutasi maka semakin cepat solusi ditemukan.

SIMPULAN

Berdasarkan penelitian yang dilakukan, didapatkan kesimpulan:

1. Model representasi kromosom yang digunakan pada penelitian ini dapat digunakan untuk mencari solusi dari permasalahan penjadwalan ujian pendadaran proyek tugas akhir.
2. Hasil pengujian menunjukkan bahwa sistem GA pada umumnya tidak dapat menemukan solusi kasus penjadwalan hingga generasi ke-10000. Sedangkan sistem *RR* mampu menemukan 2 solusi dengan rata-rata waktu eksekusi 2.30 detik saat nilai $n = 7$.
3. Terdapat pengaruh antara frekuensi mutasi dengan jumlah generasi untuk mencari solusi pada sistem *RR*. Hal tersebut dapat dilihat dari terus bertambahnya jumlah generasi yang dibutuhkan untuk mencari solusi seiring dengan semakin besarnya nilai n .
4. Pada pencarian solusi penjadwalan menggunakan konsep *RR*, nilai pengatur frekuensi mutasi n dapat dibuat kecil (mendekati 0) jika ingin mendapatkan solusi dengan cepat.

UCAPAN TERIMA KASIH

Ucapan terimakasih ditujukan kepada :

1. Kementerian Ristekdikti yang memberikan dana pada penelitian ini melalui hibah Penelitian Dosen Pemula.
2. Rektor Universitas Teknologi Yogyakarta yang sudah memberikan dukungan sehingga penelitian ini terlaksana secara baik.

DAFTAR PUSTAKA

- Borna, K. and Hashemi, V. H. (2014) 'An Improved Genetic Algorithm With A Local Optimization Strategy and an Extra Mutation Level for Solving Traveling Salesman Problem', *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, 4(4), pp. 47–53. doi: 10.5121/ijceit.2014.4405.
- Brezulianu, A., Fira, L. and Fira, M. (2012) 'A genetic algorithm approach for scheduling of resources in well-services companies', *IJARAI - International Journal of Advanced Research in Artificial Intelligence*, 1(5), pp. 1–6.
- Burke, E. K. and Petrovic, S. (2002) 'Recent Research Directions in Automated Timetabling', *European Journal of Operational Research*, 140(2), pp. 266–280.
- Johar, A., Jain, S. S. and Garg, P. K. (2016) 'Transit network design and scheduling using genetic algorithm – a review', *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 6(1), pp. 9–22. doi: 10.11121/ijocta.01.2016.00258.
- Kim, S. *et al.* (2014) 'A Strategy to Improve Performance of Genetic Algorithm for Nurse Scheduling Problem', *International Journal of Software Engineering and Its Applications*, 8(1), pp. 53–62. doi: 10.14257.
- Li, Y. and Chen, Y. (2010) 'A genetic algorithm for job-shop scheduling', *Journal of Software*, 5(3), pp. 269–274. doi: 10.4304/jsw.5.3.269-274.
- Mateescu, G. D. (2010) 'On the Implementation and Use of a Genetic Algorithm with Genetic Acquisitions', *Romanian Journal of Economic Forecasting*, 13(2), pp. 223–230. doi: 10.1093/ajae/aat038.
- Mawaddah, N. K. and Mahmudy, W. F. (2006) 'Optimasi Penjadwalan Perawat Menggunakan Algoritme Genetika', *Universitas Brawijaya*, 2(2), pp. 1–8. doi: 10.1007/s13398-014-0173-7.2.
- Nazif, H. (2012) 'A Genetic Algorithm for Solving Scheduling Problem', *The Journal of Mathematics and Computer Science*, 5(2), pp. 91–96.
- Obaid, O. I. *et al.* (2012) 'Comparing Performance of Genetic Algorithm with Varying Crossover in Solving Examination Timetabling Problem', *Journal of Emerging Trends in Computing and Information Sciences*, 3(10), pp. 1427–1434.
- Puspaningrum, W. A., Djunaidy, A. and Vinarti, R. A. (2013) 'Penjadwalan Mata Kuliah Menggunakan Algoritme Genetika di Jurusan Sistem Informasi ITS', *Jurnal Teknik Pomits*, 2(1), pp. 127–131.

- Saryanti, I. G. A. D. and Wijanegara, I. K. (2017) 'Penerapan Metode Algoritme Genetika Untuk Penjadwalan Mengajar', *Jurnal SIMETRIS*, 8(1), pp. 53–60.
- Sharma, A., Patani, R. and Aggarwal, A. (2016) 'Software Testing Using Genetic Algorithms', *International Journal of Computer Science & Engineering Survey*, 7(2), pp. 21–33. doi: 10.5121/ijcses.2016.7203.
- Suhartono, E. (2015) 'DENGAN ALGORITME GENETIKA (Studi Kasus di AMIK JTC Semarang)', *Infokam*, 2, pp. 132–146.
- Szénási, S. and Vámosy, Z. (2013) 'Implementation of a distributed genetic algorithm for parameter optimization in a cell nuclei detection project', *Acta Polytechnica Hungarica*, 10(4), pp. 59–86. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84881598860&partnerID=40&md5=387062235d2b12160572b607bc0e7d63>.
- Tabassum, M. (2014) 'a Genetic Algorithm Analysis Towards Optimization Solutions', *International Journal of Digital Information and Wireless Communications*, 4(1), pp. 124–142. doi: 10.17781/P001091.
- Taiwo, O. S., Mayowa, O. O. and Ruka, K. B. (2013) 'Application of Genetic Algorithm to Solve Traveling Salesman Problem', *International Journal of Advance Research, IJOAR*, 1(4), pp. 27–46.