



Artikel Penelitian

Pengembangan Algoritma Genetika dengan Pendekatan *Repetitive Random* untuk Penjadwalan Ujian Pendadaran Proyek Tugas Akhir

Adityo Permana Wibowo^{a*}, Donny Avianto^b, Imantoko^c

^aProgram Studi Sistem Informasi, Universitas Teknologi Yogyakarta, Jl. Siliwangi, Sleman, 55285, Indonesia

^{b,c}Program Studi Informatika, Universitas Teknologi Yogyakarta, Jl. Siliwangi, Sleman, 55285, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 15 Mei 2020

Revisi Akhir: 05 Juni 2021

Diterbitkan Online: 07 Juni 2021

KATA KUNCI

Algoritma Genetika,

Automatically Generated Scheduling,

Proyek Tugas Akhir,

Ujian Pendadaran.

KORESPONDENSI

E-mail: adityopw@uty.ac.id*

ABSTRACT

Penjadwalan adalah hal yang umum dilakukan terutama di instansi pendidikan seperti perguruan tinggi. Salah satu kegiatan yang memerlukan proses penjadwalan di perguruan tinggi adalah penjadwalan ujian pendadaran proyek tugas akhir. Faktor-faktor seperti jumlah mahasiswa, ketersediaan dosen, dan ketersediaan ruangan juga membuat proses penjadwalan menjadi lebih kompleks dan memakan waktu jika dilakukan secara manual. Metode *Genetic Algorithm* (GA) dapat dimanfaatkan untuk mengatasi masalah di dalam proses penjadwalan secara manual. Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi yang diberi nama *Automatically Generated Scheduling* (ALGEN Scheduling) yang mampu melakukan proses penjadwalan ujian pendadaran dengan lebih mudah. "ALGEN Scheduling" dibangun menggunakan bahasa C# dan menerapkan GA yang telah dimodifikasi dengan pendekatan *repetitive random* sebagai inti untuk melakukan proses penjadwalan. Pembuatan ALGEN Scheduling dimulai dari perancangan kromosom, implementasi GA termodifikasi dalam bahasa C#, pembuatan GUI dengan *tools* visual studio, dan integrasi inti penjadwalan dengan GUI. Aplikasi yang dibangun bisa membantu mahasiswa dan dosen penguji dalam menginfokan jadwal ujian pendadaran. Untuk mengetahui kinerja Aplikasi "ALGEN Scheduling", peneliti melakukan evaluasi dengan menyebar kuesioner kepada responden yang terdiri dari mahasiswa dan dosen. Peneliti berhasil mendapatkan tanggapan sebanyak 101 responden. Berdasarkan pengolahan data responden didapatkan 68% menilai bahwa Aplikasi "ALGEN Scheduling" memiliki tampilan yang bagus dan sangat mudah digunakan untuk menghasilkan jadwal ujian pendadaran dengan tanpa adanya bentrokan jadwal. Sedangkan 19% responden mengatakan bahwa Aplikasi "ALGEN Scheduling" tidak terlalu mudah digunakan tetapi juga tidak terlalu sulit untuk dipelajari. Sisanya, responden menganggap perlu adanya pelatihan khusus sebelum aplikasi benar-benar diterapkan.

1. PENDAHULUAN

Ujian pendadaran proyek tugas akhir adalah fase yang biasanya dilalui seorang mahasiswa sebelum mendapatkan gelar akademik untuk jenjang Strata-1 (S-1). Saat ujian 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 pembuatan jadwal dosen penguji ujian pendadaran yang dilakukan oleh sekretaris program studi. Jadwal yang dibuat oleh sekretaris prodi bisa menjadi kompleks karena beberapa faktor.

Faktor pertama, harus memperhatikan jadwal kegiatan dosen (jadwal mengajar) yang akan bergabung sebagai tim penguji. Tidak jarang juga sekretaris prodi harus melakukan komunikasi secara intens melalui telepon atau pesan singkat untuk mengetahui jadwal kegiatan seorang dosen. Kompleksitas pembuatan jadwal juga akan bertambah seiring dengan bertambahnya jumlah dosen yang akan bergabung sebagai tim penguji.

Faktor kedua, terbatasnya jumlah ruang dan sesi yang tersedia untuk menyelenggarakan ujian pendadaran proyek tugas akhir. Seringkali dijumpai pada suatu institusi akademik, satu ruangan digunakan untuk menyelenggarakan ujian pendadaran dari program studi atau fakultas yang berbeda-beda. Hal ini mengharuskan sekretaris prodi berkoordinasi dengan pihak-pihak lain yang memiliki hak menggunakan ruangan tersebut. Jika sekretaris prodi tidak

memperhatikan kedua faktor ini, maka ujian pendadaran harus dibatalkan dan *direshedule* karena adanya bentrok dengan jadwal lainnya. Dengan demikian membutuhkan waktu lagi untuk revisi jadwal pendadaran, serta memberikan dampak pada lamanya waktu tunggu mahasiswa untuk mendapatkan jadwal ujian pendadaran. Hal ini dapat dihindari melalui penggunaan aplikasi penjadwalan karena waktu yang dibutuhkan untuk melakukan verifikasi jadwal kegiatan dosen dan melihat ketersediaan jumlah ruang tiap sesi untuk melakukan penjadwalan dapat dikurangi sehingga mahasiswa bisa mendapatkan jadwal ujian pendadaran lebih cepat.

Berdasarkan permasalahan yang sudah dijabarkan, penulis mencoba untuk membuat sebuah aplikasi untuk penjadwalan dosen penguji ujian pendadaran menggunakan metode Algoritma Genetika atau *Genetic Algorithm* (GA). Metode GA merupakan salah satu metode yang prosesnya menggunakan konsep perilaku evolusi biologis untuk memecahkan suatu masalah tertentu secara terprogram [1]. Di Indonesia sendiri sudah banyak penggunaan metode GA untuk membuat jadwal di suatu institusi pendidikan maupun institusi non pendidikan.

Penggunaan GA untuk pemodelan jadwal dimulai dari penjabaran *constraint* yang digunakan. *Constraint* tersebut nantinya dikonversi menjadi gen yang akan digunakan untuk proses evolusi. Setelah perhitungan evolusi, selanjutnya dilakukan proses perhitungan untuk mencari nilai *fitness* yang terbesar. Jika proses tersebut dilakukan dengan benar dan direpresentasikan dengan jelas, maka akan menghasilkan model GA yang terbaik dan menghasilkan aplikasi penjadwalan yang optimal [2]. Namun sebaliknya jika tidak direpresentasikan dengan jelas maka bisa menghasilkan model yang tidak maksimal [3] [4].

Tujuan yang akan dicapai pada penelitian ini adalah mengembangkan sebuah aplikasi penjadwalan dosen penguji ujian pendadaran proyek tugas akhir menggunakan Algoritma Genetika yang sudah dimodifikasi pada bagian pencarian nilai *fitness* dengan pendekatan *repetitive random* sehingga proses *generate* jadwal menjadi lebih cepat dan mnghasilkan jadwal yang optimal [5].

Beberapa penelitian yang menunjukkan kemampuan GA untuk menemukan solusi yang optimal di berbagai bidang sudah banyak dibuktikan, seperti: penelitian di bidang distribusi logistik untuk menyelesaikan permasalahan *Travelling Salesman Problem* [6] dan [7], di bidang biologi untuk optimalisasi parameter dalam deteksi inti sel [8], di bidang rekayasa perangkat lunak untuk meakukan pengujian perangkat lunak [9], dan di bidang ekonomi untuk menentukan parameter pada fungsi produksi Cobb-Douglas [10].

Selain digunakan untuk mencari solusi dari permasalahan optimalisasi, penelitian sebelumnya juga menunjukkan penggunaan GA untuk menangani masalah penjadwalan. Seperti penelitian tentang penjadwalan penggunaan sumber daya pada perusahaan *well-service* [11], jadwal transit untuk kendaraan umum [12], jadwal jaga perawat di rumah sakit [13] dan jadwal penggunaan mesin di pabrik Li dan Chen [14] dan [15]. Penelitian-penelitian tersebut menunjukkan bahwa GA mampu menangani permasalahan

penjadwalan karena pada dasarnya penjadwalan juga merupakan masalah optimasi yang dapat diselesaikan dengan cara *generate* beberapa alternatif solusi secara iteratif menggunakan suatu fungsi *fitness* tertentu.

1.1. Algoritma Genetika

Algoritma Genetika merupakan suatu algoritma pencarian yang didasarkan pada konsep terbentuknya populasi secara acak dari beberapa individu dan sifatnya tergantung pada gen yang terdapat pada kromosomnya. Individu melakukan proses reproduksi untuk melahirkan keturunan, sifat keturunan dibentuk dari kombinasi sifat kedua induknya atau mewarisi sifat induknya [16].

Konsep tersebut membuat Algoritma Genetika menjadi algoritma yang tepat jika digunakan untuk penyelesaian masalah optimasi yang kompleks yang tidak bisa dipecahkan oleh metode konvensional [16].

1.2. Skema Pengkodean

Suatu permasalahan jika akan diselesaikan menggunakan Algoritma Genetika, harus dikonversi terlebih dulu ke dalam bentuk individu yang diawali oleh satu atau lebih kromosom dengan kode tertentu. Berbeda dengan teori genetika di dunia nyata yang merepresentasikan gen sebagai deretan *based*. Algoritma Genetika merepresentasikan gen (buatan), secara umum, sebagai bilangan real, desimal atau biner. Skema pengkodean Algoritma Genetika ada tiga macam, yaitu [17]:

- Real-number encoding**, pada skema ini, nilai gen berada dalam interval $[0, R]$, dimana R adalah bilangan real positif dan biasanya $R = 1$.
- Discrete decimal encoding**, pada skema ini, setiap gen bisa berupa deretan bilangan bulat dalam interval $[0, 9]$.
- Binary encoding**, setiap gen bisa berupa deretan nilai 0 atau 1.

1.3. Nilai Fitness

Teori evolusi di dunia nyata, individu yang mempunyai nilai *fitness* (kebugaran) tinggi akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati. Algoritma Genetika (GA) menerangkan bahwa sebuah individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*-nya. Pada kasus optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi h (dikenal sebagai masalah maksimasi), maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f = h$ (dimana f adalah nilai *fitness*) tetapi jika masalahnya adalah meminimalkan fungsi h (masalah minimasi) maka fungsi h tidak bisa digunakan secara langsung [17]. Oleh karena itu, nilai *fitness* yang digunakan untuk memecahkan permasalahan minimasi adalah $f = 1/h$, artinya semakin kecil nilai h semakin besar nilai f . Tetapi, fungsi ini akan bermasalah jika h bisa bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasi masalah tersebut, h perlu ditambah dengan sebuah bilangan yang dianggap sangat kecil, sehingga formula fungsi *fitness*-nya menjadi seperti persamaan 2.1.

$$f = \frac{1}{(h+a)}, \quad (2.1)$$

Keterangan:

f : Nilai *Fitness*

h : Masalah Minimasi

a : Nilai Variabel (a,b,c,...n)

Dimana a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan ketentuan masalah yang akan diselesaikan. Misalnya, jika diketahui bahwa $x_1, x_2 \in [-4, +4]$, maka formula fungsi *fitness* yang bisa digunakan untuk mencari nilai minimal dari fungsi tersebut diolah menggunakan persamaan 2.2.

$$h(x_1, x_2) = 5x_1^4 + 4x_2^2. \quad (2.2)$$

Keterangan:

h : Masalah Minimasi

x_1, x_2 : Nilai Variabel

Fungsi h mencapai nilai minimum 0, pada saat $x_1 = 0$ dan $x_2 = 0$. karena h bisa bernilai 0, maka nilai *fitness* yang bisa digunakan untuk masalah ini adalah $f = 1/(h+a)$. jika mendefinisikan a sebagai 0.001, maka nilai maksimum f adalah 1000 (ketika $h = 0$).

Terkadang kita menghadapi masalah dimana fungsi h memiliki variasi sangat kecil, bagaimana formula fungsi *fitness* yang sebaiknya kita gunakan untuk menemukan x_1 dan x_2 yang memaksimalkan persamaan 2.3 (jika diketahui $x_1, x_2 \in [-2, +2]$)?

$$h(x_1, x_2) = 100000 + 5x_1 + x_2 \quad (2.3)$$

Pada fungsi persamaan 2.3, nilai-nilai h berada dalam interval 99988 sampai 100012, dengan demikian semua individu memiliki nilai *fitness* yang hampir sama dalam kisaran 100000. Hal ini akan berakibat buruk pada proses seleksi pemilihan orang tua secara proporsional sesuai dengan nilai *fitness*-nya untuk itu diperlukan suatu mekanisme yang disebut **linear fitness ranking**, mekanisme ini bertujuan untuk melakukan pen-skala-an nilai-nilai *fitness*, individu yang mempunyai nilai *fitness* tertinggi ke dua diberi *fitness* $N-1$ dan seterusnya sehingga individu yang mempunyai nilai *fitness* terendah diberi nilai *fitness* 1. Misalnya $R(i)$ menyatakan ranking individu ke i , $R(i) = 1$ jika i adalah individu bernilai *fitness* tertinggi dan $R(i) = N$ jika i adalah individu bernilai *fitness* terendah, dengan demikian, formula fungsi *fitness* yang sebaiknya digunakan untuk masalah tersebut adalah seperti persamaan 2.4

$$f(i) = (N+1 - R(i)) \quad (2.4)$$

Keterangan:

$f(i)$: Fungsi Nilai *Fitness*

N : Ukuran Populasi

R : Iterasi ke-1

Penggunaan formula fungsi *fitness* pada persamaan 2.4 di atas, bisa berakibat evolusi akan mencapai optimum lokal karena kecilnya variasi nilai *fitness* pada semua individu dalam populasi. Kecenderungan untuk konvergen pada optimum lokal dapat dikurangi dengan menggunakan formula pada persamaan 2.5 di bawah ini. Formula tersebut akan menghasilkan nilai *fitness* yang berada dalam interval $[f_{\min}, f_{\max}]$, dimana f_{\max} dan f_{\min} , secara berturut-turut, adalah nilai *fitness* terkecil dan terbesar.

$$f(i) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(i)-1}{N-1} \right) \quad (2.5)$$

Keterangan:

f : Fungsi Nilai *Fitness*

f_{\max} : *Fitness* Maksimum

f_{\min} : *Fitness* Minimum

$R(i)$: Iterasi ke-1

$N-1$: Ukuran Populasi

Penentuan nilai *fitness* sangat berpengaruh pada performansi Algoritma Genetika. Permasalahan sederhana seperti optimasi fungsi dan formula fungsi *fitness* bisa ditemukan dengan mudah. Tetapi, pada umumnya kebanyakan permasalahan kompleks, seperti penjadwalan, peramalan, *learning*, dan sebagainya, diperlukan usaha cukup berat untuk menemukan formula fungsi *fitness* yang baik.

2. 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, serta representasi gen yang digunakan.

2.1. Kendala-kendala (Constraints) yang digunakan dalam proses Algoritma Genetika

Terdapat dua *constraint*/batasan dalam penyusunan penjadwalan yaitu *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi) [18]. *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 [19].

Penelitian ini menggunakan *constraint* yang dimaksudkan agar model bisa menghasilkan sistem yang sesuai dengan peraturan yang

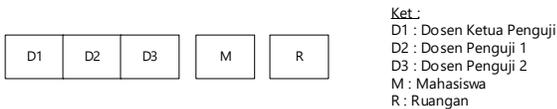
sudah ditetapkan di Program Studi Teknik Informatika Universitas Teknologi Yogyakarta sebelumnya. *Constraint* ini juga digunakan untuk menentukan nilai *fitness* untuk perhitungan algoritma genetika. Adapun poin-poin yang penulis tetapkan sebagai *constraint* pada penelitian ini adalah sebagai berikut :

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 Algoritma Genetika. Terdapat 20 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.

2.2. Representasi Gen

Berdasarkan teknis pelaksanaan ujian pendadaran yang sudah dijelaskan, maka komponen Algoritma 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.



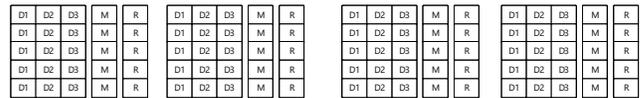
Gambar 1. Gen yang membentuk 1 subkromosom

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



Gambar 2. Subkromosom yang membentuk 1 kromosom

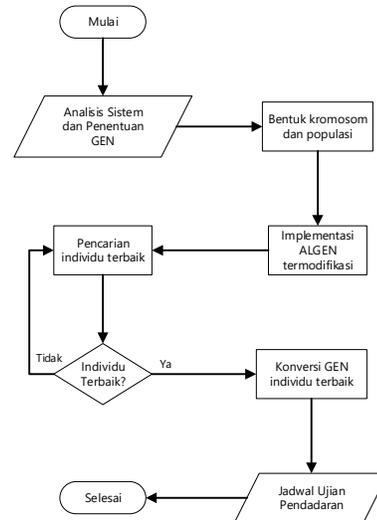
Pada penelitian ini, untuk membentuk sebuah individu, digunakan pengelompokkan sebanyak lima kromosom. Individu yang terbentuk untuk penelitian ini, seperti terlihat pada Gambar 3.



Gambar 3. Kromosom yang membentuk 1 Individu

2.3. Alur Sistem aplikasi yang diusulkan

Secara garis besar pembuatan Aplikasi ALGEN Scheduling, dapat dibagi ke dalam enam tahap. Tahapan pertama adalah melakukan analisa sistem pembuatan jadwal ujian pendadaran tugas akhir yang sedang digunakan. Tahapan kedua, merepresentasikan gen ke dalam bentuk kromosom dan populasi yang sesuai dengan permasalahan yang akan diselesaikan. Tahapan ketiga adalah mengaplikasikan algoritma genetika yang sudah dimodifikasi untuk memproses populasi yang terbentuk. Tahapan keempat, sistem kemudian akan melakukan pencarian individu terbaik melalui proses yang iteratif. Setelah individu terbaik ditemukan, tahapan kelima adalah melakukan konversi gen individu terbaik. Konversi gen individu terbaik dilakukan dengan tujuan mengubah tampilan gen milik individu terbaik yang awalnya berisi angka-angka menjadi bentuk tabel yang berisi jadwal pendadaran sehingga hasil dari algoritma genetika dapat lebih mudah dibaca dan dipahami. Tahap terakhir adalah melakukan evaluasi terhadap tampilan hasil konversi gen individu. Adapun tahapan pembuatan aplikasi dapat dilihat pada Gambar 4 di bawah ini.

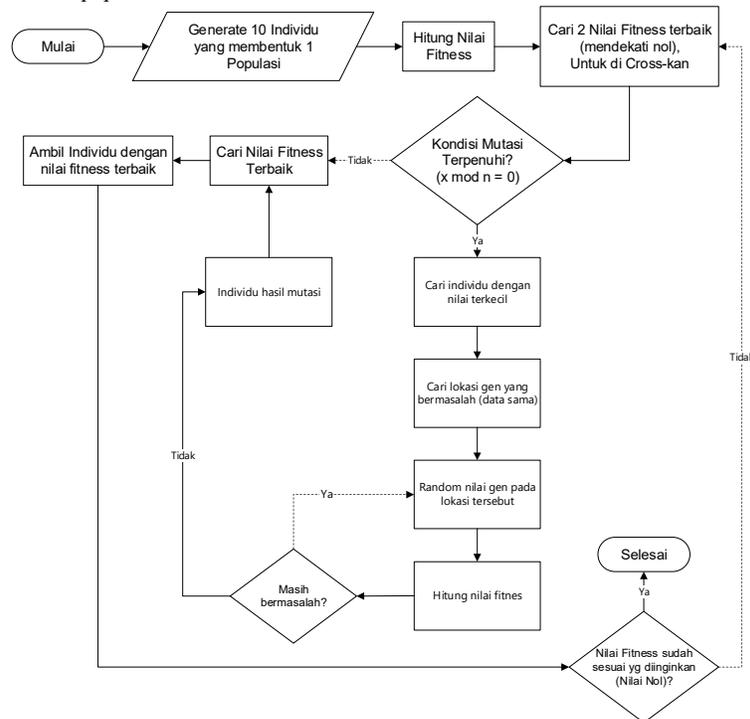


Gambar 4. Proses Pembuatan Aplikasi ALGEN Scheduling

Adapun modifikasi Algoritma Genetika yang dimaksud pada tahap ketiga Gambar 4 terletak pada proses mutasi. Algoritma Genetika pada umumnya akan menghitung ulang semua gen jika nilai *fitness* nya tidak bernilai 0. Proses tersebut tentunya akan membutuhkan waktu lama untuk proses *generate* penjadwalan. Untuk mempersingkat waktu dalam proses *generate* jadwal, peneliti mencoba memodifikasi proses perhitungan nilai fitnessnya dengan mencari gen mana yang bermasalah sehingga gen yang tidak bermasalah akan diabaikan. Gen yang bermasalah akan dihitung ulang nilai fitnessnya sampai dengan bernilai 0. Proses tersebut diberi nama *Repetitive Random* [5].

Alur sistem yang dibuat dimulai dari 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.

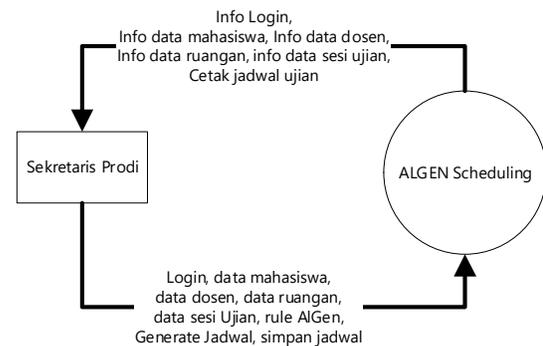
Selanjutnya sistem akan mengevaluasi nilai $x \bmod n$, dimana x adalah bilangan bulat hasil random dan n adalah bilangan bulat yang menentukan frekuensi mutasi sistem, maka mutasi akan dilakukan. Terakhir, sistem akan menghitung lagi nilai *fitness* setiap individu dan jika ada individu dengan nilai *fitness* = 0 maka sistem akan berhenti, namun jika tidak, maka sistem mengulang proses. Alur Algoritma Genetika secara umum terlihat pada Gambar 5.



Gambar 5. Flowchart pencarian komposisi terbaik pada Algoritma Genetika

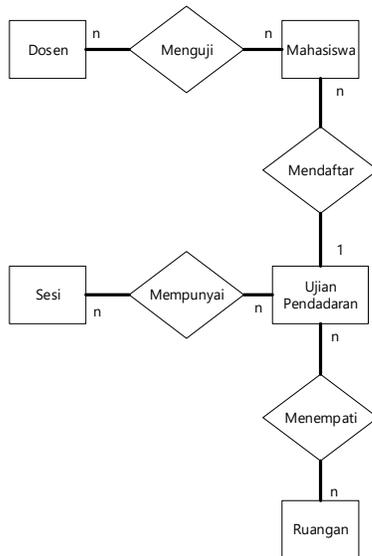
2.4. Perancangan Aplikasi ALGEN Scheduling

Perancangan aplikasi ALGEN Scheduling digambarkan dengan Diagram Alur Data (DAD) sebagai gambaran proses yang terjadi. Sedangkan rancangan basis datanya menggunakan Entity Relationship Diagram (ERD). Penggambaran DAD terdiri dari 1 proses dan 1 entitas dimana entitasnya adalah pengguna aplikasi. Pengguna aplikasi ALGEN Scheduling adalah Sekretaris Program Studi Teknik Informatika, karena tugas pembuatan jadwal ujian pendarasan dilakukan oleh sekretaris prodi. Hak akses yang bisa dilakukan oleh sekretaris prodi adalah menginputkan data-data yang digunakan untuk proses penjadwalan. Seperti terlihat pada Gambar 6.



Gambar 6. Diagram Alur Data Aplikasi ALGEN Scheduling

Setelah didapatkan rancangan proses aplikasi ALGEN Scheduling, didapatkan alur data yang mengalir dari dan menuju proses aplikasi tersebut. Alur data tersebut menunjukkan bahwa data yang akan diproses dan disimpan dalam basis data. Perancangan basis data aplikasi ALGEN Scheduling seperti terlihat pada Gambar 7.



Gambar 7. ERD Aplikasi ALGEN Scheduling

2.5. Pengujian Sistem

Pengujian sistem difokuskan pada tampilan dan fungsionalitas aplikasi dalam memproses penjadwalan ujian pendadaran. Pengujian melibatkan 101 responden. Mula-mula setiap responden diminta untuk mencoba aplikasi yang dibuat, kemudian responden akan diminta menjawab 8 butir pertanyaan yang berhubungan dengan aplikasi tersebut.

Ke-8 butir pertanyaan tersebut antara lain, butir 1 berisi tentang penilaian tampilan aplikasi secara keseluruhan. Butir 2 berisi tentang penggunaan aplikasi. Butir 3 berisi tentang pewarnaan halaman tampilan aplikasi. Butir 4 berisi tentang kekurangan aplikasi. Butir 5 berisi tentang tanggapan pengguna terkait jadwal yang dihasilkan oleh aplikasi. Butir 6 berisi relevansi jadwal yang dihasilkan oleh aplikasi dengan kondisi yang sebenarnya. Butir 7 berisi masukan dari pengguna terkait aplikasi penjadwalan. Butir 8 berisi poin atau bagian yang perlu diubah dari aplikasi yang dibuat.

Setelah seluruh responden menjawab pertanyaan dikuesioner, Langkah berikutnya adalah menghitung jumlah jawaban dari setiap pertanyaan dan membuat persentasenya. Persentase ini yang digunakan untuk menilai aplikasi secara keseluruhan.

3. HASIL

Pada bagian ini, akan dibahas hasil penelitian berupa aplikasi penjadwalan yang sudah dibuat menggunakan metode yang sudah dijabarkan pada bagian sebelumnya. Penelitian sebelumnya yang pernah dilakukan oleh [5], bahwa aplikasi penjadwalan yang dibuat masih berbasis konsol dan berupa tampilan kode-kode sebagai inisialisasi dari *gen*. Sedangkan Aplikasi penjadwalan yang dibuat

pada penelitian ini sudah berbasis GUI (*Graphic User Interface*), aplikasi tersebut bernama “ALGEN Scheduling”.

Halaman utama aplikasi ALGEN Scheduling terdapat logo aplikasi pada bagian utama, kemudian terdapat 6 menu pada bagian sebelah kiri. Menu tersebut terdiri dari Beranda, input data Dosen yang bertindak sebagai dosen pembimbing dan penguji, input data Mahasiswa yang mendaftar ujian pendadaran, input data ruangan yang digunakan untuk ujian pendadaran, tampilan proses jadwal ujian pendadaran, serta tampilan halaman tentang. Halaman utama aplikasi penjadwalan ujian pendadaran menggunakan Algoritma Genetika terlihat pada Gambar 8.



Gambar 8. Halaman Utama Aplikasi ALGEN Scheduling

Penggunaan aplikasi dimulai dari mengisi data dosen penguji ujian pendadaran. Sebenarnya data dosen yang terdapat pada sistem ini terdiri dari dosen pembimbing proyek tugas akhir dan dosen penguji proyek tugas akhir. Hanya saja ketika ujian pendadaran, dosen pembimbing bisa menjadi dosen penguji. Data dosen yang diinputkan terdiri dari NIK, nama dosen, nomor telepon dosen, alamat dosen dan foto dosen. Input data melalui komponen *edit text* yang sudah disediakan. Data dosen yang sudah diinputkan akan tampil pada bagian kanan dari inputan data. Tampilan halaman input dosen penguji terlihat pada Gambar 9.



Gambar 9. Halaman Input data Dosen Penguji

Selain input data dosen penguji, aplikasi juga bisa memproses perubahan data dosen penguji yang sudah diinputkan. Halaman ubah data dosen muncul ketika diklik tampilan data/foto pada halaman data dosen penguji. Tampilan halaman ubah data dosen penguji terlihat pada Gambar 10.

Gambar 10. Halaman Ubah Data Dosen Penguji

Selain menginputkan dan menampilkan data dosen, aplikasi ALGEN Scheduling juga terdapat halaman input data mahasiswa dan judul proyek tugas akhir yang dilakukan. Data mahasiswa yang diinputkan terdiri dari NIM, nama, judul tugas akhir, nomor telepon, alamat dan foto mahasiswa. Tampilan halaman input data mahasiswa terlihat pada Gambar 11.

Gambar 11. Halaman Input Data Mahasiswa

Selain input data mahasiswa, aplikasi juga bisa memproses perubahan data mahasiswa yang sudah diinputkan. Halaman ubah data mahasiswa muncul ketika diklik tampilan data/foto pada halaman data mahasiswa. Tampilan halaman ubah data mahasiswa terlihat pada Gambar 12.

Gambar 12. Halaman Ubah Data Mahasiswa

Selain input data dosen dan data mahasiswa, juga ada halaman tampilan input data ruangan untuk ujian pendadaran. Dalam hal ini ruangan yang digunakan hanya 2, namun jika dikemudian hari

ruangannya bertambah, bisa ditambahkan pada aplikasi. Tampilan halaman input ruangan terlihat pada Gambar 13.

Gambar 13. Halaman Input Data Ruangan

Jika data master sudah diinputkan pada halaman di atas, selanjutnya ke menu Jadwal. Pada tampilan menu jadwal terdapat 3 *button*, yaitu *button* proses, cetak, dan bersihkan. Tampilan halaman menu untuk proses penjadwalan terlihat pada Gambar 14.

Gambar 14. Halaman Penjadwalan sebelum dieksekusi

Button proses digunakan untuk memproses penjadwalan ujian pendadaran menggunakan Algoritma Genetika. Setelah klik *button* proses, maka aplikasi “ALGEN Scheduling” akan mulai menjalankan proses penjadwalan. Waktu proses yang terjadi tergantung jumlah mahasiswa yang melakukan pendaftaran ujian pendadaran. Untuk mengetahui keberlangsungan proses pada aplikasi “ALGEN Scheduling”, terdapat progress bar yang menunjukkan prosentase proses penjadwalan. Tampilan progress bar untuk penjadwalan terlihat pada Gambar 15.

Gambar 15. Tampilan *Progress Bar* Proses Penjadwalan

Jika progress bar menunjukkan angka 100%, artinya proses *generate* jadwal sudah selesai, selanjutnya akan tampil daftar jadwal

ujian pendadaran hasil *generate* Algoritma Genetika. Tampilan penjadwalan hasil *generate* program terlihat pada Gambar 16.

Hari	Ruangan	Sesi	Mahasiswa	Pembimbing	Ketua Penguji	Penguji
Senin	Ruangan 1	08.00 - 10.00	Apri Yanto	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.	Rianto, S.Kom., M.Eng.
		11.00 - 13.00	Resky Nuseptadi	Yuli Asningtias, S.Kom., M.Kom.	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.
	Ruangan 2	08.00 - 10.00	Febri Saputra	Dr. Ir. Anief Hemawan, S.T., M.T.	Yuli Asningtias, S.Kom., M.Kom.	Donny Avianto, S.T., M.T.
		11.00 - 13.00	Yusiana Syafiq Azhar	Sutaman, S.Kom., M.Kom., Ph.D.	Donny Avianto, S.T., M.T.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.
Selasa	Ruangan 1	08.00 - 10.00	Rio Ardiansyah	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Sutaman, S.Kom., M.Kom., Ph.D.	Suhiman, S.Kom., M.Kom., Ph.D.
		11.00 - 13.00	Tegar Adi Wicaksono	Yuli Asningtias, S.Kom., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.
	Ruangan 2	08.00 - 10.00	Reudy Eka Setiady	Dr. Ir. Anief Hemawan, S.T., M.T.	Adityo Permana Wibowo, S.Kom., M.Cs.	Rianto, S.Kom., M.Eng.
		11.00 - 13.00	Suptyanto	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Adityo Permana Wibowo, S.Kom., M.Cs.	Iwan Hartadi Tri Untoro, S.T., M.Kom.
Rabu	Ruangan 1	08.00 - 10.00	Agus Andi Santoso	Sutaman, S.Kom., M.Kom., Ph.D.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.
		11.00 - 13.00	Handianus	Donny Avianto, S.T., M.T.	Rianto, S.Kom., M.Eng.	Suhiman, S.Kom., M.Kom., Ph.D.
	Ruangan 2	08.00 - 10.00	Yusuf Izza	Dr. Ir. Anief Hemawan, S.T., M.T.	Rianto, S.Kom., M.Eng.	Suhiman, S.Kom., M.Kom.
		11.00 - 13.00	Pipit Ari Mufidah	Yuli Asningtias, S.Kom., M.Kom.	Adityo Permana Wibowo, S.Kom., M.Cs.	Dr. Ir. Anief Hemawan, S.T., M.T.
Kamis	Ruangan 1	08.00 - 10.00	Mabakbul Munir	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.	Donny Avianto, S.T., M.T.
		11.00 - 13.00	Ahmad Mutadho	Rianto, S.Kom., M.Eng.	Sutaman, S.Kom., M.Kom., Ph.D.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.
	Ruangan 2	08.00 - 10.00	Muhlis Aptyan	Sutaman, S.Kom., M.Kom., Ph.D.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Suhiman, S.Kom., M.Kom., Ph.D.
		11.00 - 13.00	Christa Bayu Aji	Dr. Ir. Anief Hemawan, S.T., M.T.	Dr. Enny Ije Sela, S.Si., M.Kom.	Donny Avianto, S.T., M.T.
Jumat	Ruangan 1	08.00 - 10.00	Muchammad Dawam Ma'arif	Dr. Ir. Anief Hemawan, S.T., M.T.	Donny Avianto, S.T., M.T.	Adityo Permana Wibowo, S.Kom., M.Cs.
		11.00 - 13.00	Ichsan Ananto	Adityo Permana Wibowo, S.Kom., M.Cs.	Rianto, S.Kom., M.Eng.	Donny Avianto, S.T., M.T.
	Ruangan 2	08.00 - 10.00	Ahmad Muallim	Suhiman, S.Kom., M.Kom., Ph.D.	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.
		11.00 - 13.00	Marsodik	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Dr. Ir. Anief Hemawan, S.T., M.T.

Gambar 16. Halaman Penjadwalan hasil *generate* menggunakan Algoritma Genetika

Setelah muncul jadwal hasil *generate*, selanjutnya bisa dilakukan proses cetak jadwal yang selanjutnya akan diumumkan ke mahasiswa dan dosen yang bersangkutan. Hasil cetak jadwal terlihat pada Gambar 17.

Hari	Ruang	Sesi	Mahasiswa	Pembimbing	Ketua Penguji	Penguji
Senin	Ruang 1	08.00 - 10.00	Apri Yanto	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.	Rianto, S.Kom., M.Eng.
		11.00 - 13.00	Resky Nuseptadi	Yuli Asningtias, S.Kom., M.Kom.	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.
Selasa	Ruang 1	08.00 - 10.00	Rio Ardiansyah	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Sutaman, S.Kom., M.Kom., Ph.D.	Suhiman, S.Kom., M.Kom., Ph.D.
		11.00 - 13.00	Tegar Adi Wicaksono	Yuli Asningtias, S.Kom., M.Kom.	Dr. Enny Ije Sela, S.Si., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.
Rabu	Ruang 1	08.00 - 10.00	Agus Andi Santoso	Sutaman, S.Kom., M.Kom., Ph.D.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.
		11.00 - 13.00	Handianus	Donny Avianto, S.T., M.T.	Rianto, S.Kom., M.Eng.	Suhiman, S.Kom., M.Kom., Ph.D.
Kamis	Ruang 1	08.00 - 10.00	Mabakbul Munir	Iwan Hartadi Tri Untoro, S.T., M.Kom.	Yuli Asningtias, S.Kom., M.Kom.	Donny Avianto, S.T., M.T.
		11.00 - 13.00	Ahmad Mutadho	Rianto, S.Kom., M.Eng.	Sutaman, S.Kom., M.Kom., Ph.D.	Dr. Erik Iman Heri Ujianto, S.T., M.Kom.
Jumat	Ruang 1	08.00 - 10.00	Muchammad Dawam Ma'arif	Dr. Ir. Anief Hemawan, S.T., M.T.	Donny Avianto, S.T., M.T.	Adityo Permana Wibowo, S.Kom., M.Cs.
		11.00 - 13.00	Ichsan Ananto	Adityo Permana Wibowo, S.Kom., M.Cs.	Rianto, S.Kom., M.Eng.	Donny Avianto, S.T., M.T.
Sabtu	Ruang 1	08.00 - 10.00	Muchammad Dawam Ma'arif	Dr. Ir. Anief Hemawan, S.T., M.T.	Donny Avianto, S.T., M.T.	Adityo Permana Wibowo, S.Kom., M.Cs.
		11.00 - 13.00	Ichsan Ananto	Adityo Permana Wibowo, S.Kom., M.Cs.	Rianto, S.Kom., M.Eng.	Donny Avianto, S.T., M.T.
Minggu	Ruang 1	08.00 - 10.00	Muchammad Dawam Ma'arif	Dr. Ir. Anief Hemawan, S.T., M.T.	Donny Avianto, S.T., M.T.	Adityo Permana Wibowo, S.Kom., M.Cs.
		11.00 - 13.00	Ichsan Ananto	Adityo Permana Wibowo, S.Kom., M.Cs.	Rianto, S.Kom., M.Eng.	Donny Avianto, S.T., M.T.

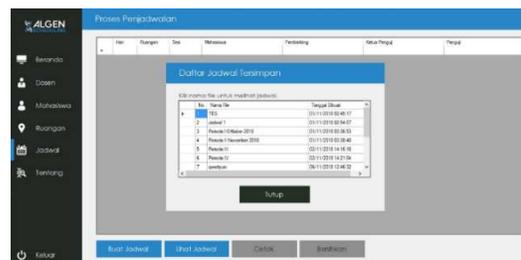
Gambar 17. Hasil Print Preview Aplikasi "ALGEN Scheduling"

Semua jadwal hasil *generate* bisa disimpan dalam basis data. Sehingga jadwal-jadwal yang pernah dibuat sebelumnya bisa ditampilkan kembali jika diperlukan. Pembuatan jadwal sesuai periode bimbingan tiap semester sehingga inisialisasi jadwal berdasarkan semester dan tahun ajaran. Tampilan halaman jadwal tersimpan terlihat pada Gambar 18.

4. PEMBAHASAN

Untuk mengetahui kinerja Aplikasi "ALGEN Scheduling", peneliti melakukan evaluasi tampilan aplikasi dengan menyebar kuesioner kepada beberapa responden yang terdiri dari dosen dan mahasiswa. Meskipun dosen dan mahasiswa tidak menggunakan aplikasi

ALGEN Scheduling secara langsung, mereka tetap dilibatkan karena informasi yang dihasilkan oleh aplikasi harus mudah dibaca dan dipahami baik oleh dosen maupun mahasiswa.



Gambar 18. Halaman Simpan Jadwal

Pada penelitian ini, peneliti berhasil mendapatkan tanggapan dari 101 responden dari kuesioner yang dibagikan. Diantara 101 responden yang mengisi kuesioner, peneliti berhasil mendapatkan respon dari Ketua Program Studi Informatika dan Wakil Dekan Fakultas Teknologi Informatika dan Elektro yang nantinya akan bersinggungan secara langsung dengan Aplikasi "ALGEN Scheduling" dalam pembuatan jadwal ujian pendadaran.

Berdasarkan hasil pengolahan data kuesioner yang dilakukan, diketahui sebanyak 68% responden menilai bahwa Aplikasi "ALGEN Scheduling" memiliki tampilan yang bagus dan sangat mudah digunakan untuk menghasilkan jadwal ujian pendadaran yang akurat tanpa adanya bentrokan jadwal. Sedangkan sekitar 19% responden mengatakan bahwa aplikasi "ALGEN Scheduling" tidak terlalu mudah digunakan tetapi juga tidak terlalu sulit untuk dipelajari. Sisanya, responden menganggap perlu adanya pelatihan khusus sebelum Aplikasi "ALGEN Scheduling" benar-benar diterapkan.

Hasil kuesioner juga menangkap harapan dari para responden terhadap Aplikasi “ALGEN Scheduling” yang dibuat. Hasil analisis menunjukkan 47% responden mengharapkan bisa menghemat waktu dan biaya serta jadwal ujian pendadaran bisa segera dipublikasikan ke mahasiswa. Disisi lain, 40% responden mengharapkan Aplikasi “ALGEN Scheduling” bisa menjadi acuan untuk pembuatan jadwal ujian pendadaran dan dapat mempertemukan tema penelitian mahasiswa dengan kompetensi dosen. Sedangkan sisanya 14% responden mengharapkan aplikasi dapat diakses oleh mahasiswa secara realtime, meskipun terdapat pembatasan fitur

5. KESIMPULAN

Berdasarkan hasil pengujian, aplikasi penjadwalan ujian pendadaran menggunakan Algoritma Genetika yang dibangun memiliki tampilan yang bagus dan mudah digunakan. Selain itu aplikasi juga sudah memenuhi harapan responden yaitu dapat menghemat waktu. Selain itu hasil pengujian menunjukkan bahwa aplikasi bisa mempertemukan antara tema penelitian mahasiswa dengan kompetensi dosen. Jika dilihat dari sisi sekretaris prodi (Sekprodi) Informatika bahwa aplikasi yang dibangun memudahkan Sekprodi dalam membuat jadwal ujian pendadaran.

Hasil dari pengujian juga mengungkapkan beberapa hal yang dapat dilakukan sebagai kelanjutan dari penelitian ini. Beberapa hal yang dapat dilakukan berikutnya antara lain: Melakukan penambahan jumlah kromosom untuk mengakomodir faktor kompetensi dosen karena faktor ini juga sering menjadi pertimbangan dalam proses penjadwalan tugas akhir mahasiswa. Menambahkan fitur untuk memeriksa kesesuaian kompetensi antara mahasiswa yang diuji dengan dosen penguji secara otomatis pada tahap *generate javascript*, mengubah jenis basis data menjadi basis data tertanam yang akan mempermudah dalam proses penyebaran aplikasi, dan penambahan fitur aplikasi suatu memodifikasi representasi gen secara dinamis sehingga pengguna aplikasi dapat menggunakan aplikasi sesuai dengan kebutuhan di instansinya.

UCAPAN TERIMA KASIH

Penelitian ini didanai oleh hibah Penelitian Dosen Pemula Kementerian Ristekdikti.

DAFTAR PUSTAKA

- [1] O. I. Obaid, M. Ahmad, S. a Mostafa, and M. A. Mohammed, “Comparing Performance of Genetic Algorithm with Varying Crossover in Solving Examination Timetabling Problem,” *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, no. 10, pp. 1427–1434, 2012.
- [2] E. Suhartono, “OPTIMASI PENJADWALAN MATAKULIAH DENGAN ALGORITMA GENETIKA (Studi Kasus di AMIK JTC Semarang),” *Infokam*, vol. 11, no. 5, pp. 132–146, 2015.
- [3] I. G. A. D. Saryanti and I. K. Wijanegara, “PENERAPAN METODE ALGORITMA GENETIKA UNTUK PENJADWALAN MENGAJAR,” *J. SIMETRIS*, vol. 8, no. 1, pp. 53–60, 2017.
- [4] W. A. Puspaningrum, A. Djunaidy, and R. A. Vinarti, “Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS,” *J. Tek. Pomits*, vol. 2, no. 1, pp. 127–131, 2013.
- [5] A. P. Wibowo and D. Avianto, “Penjadwalan Ujian Pendadaran Proyek Tugas Akhir Menggunakan Algoritma Genetika Dengan Repetitive Random Approach Final Project Examination Scheduling Using Genetic Algorithm with Repetitive Random Approach,” *Pros. Semin. Ilm. Ilmu Komput. 2018 Mach. Learn. Agric. 4.0*, pp. 61–70, 2018.
- [6] K. Borna and V. H. Hashemi, “An Improved Genetic Algorithm With A Local Optimization Strategy and an Extra Mutation Level for Solving Traveling Salesman Problem,” *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 4, no. 4, pp. 47–53, 2014.
- [7] O. S. Taiwo, O. O. Mayowa, and K. B. Ruka, “Application of Genetic Algorithm to Solve Traveling Salesman Problem,” *Int. J. Adv. Res. IJOAR*, vol. 1, no. 4, pp. 27–46, 2013.
- [8] S. Szénási and Z. Vámosy, “Implementation of a distributed genetic algorithm for parameter optimization in a cell nuclei detection project,” *Acta Polytech. Hungarica*, vol. 10, no. 4, pp. 59–86, 2013.
- [9] A. Sharma, R. Patani, and A. Aggarwal, “Software Testing Using Genetic Algorithms,” *Int. J. Comput. Sci. Eng. Surv.*, vol. 7, no. 2, pp. 21–33, 2016.
- [10] G. D. Mateescu, “On the Implementation and Use of a Genetic Algorithm with Genetic Acquisitions,” *Rom. J. Econ. Forecast.*, vol. 13, no. 2, pp. 223–230, 2010.
- [11] A. Brezulianu, L. Fira, and M. Fira, “A genetic algorithm approach for scheduling of resources in well-services companies,” *IJARAI - Int. J. Adv. Res. Artif. Intell.*, vol. 1, no. 5, pp. 1–6, 2012.
- [12] A. Johar, S. S. Jain, and P. K. Garg, “Transit network design and scheduling using genetic algorithm – a review,” *An Int. J. Optim. Control Theor. Appl.*, vol. 6, no. 1, pp. 9–22, 2016.
- [13] S. Kim, Y. Ko, S. Uhm, and J. Kim, “A Strategy to Improve Performance of Genetic Algorithm for Nurse Scheduling Problem,” *Int. J. Softw. Eng. Its Appl.*, vol. 8, no. 1, pp. 53–62, 2014.
- [14] Y. Li and Y. Chen, “A genetic algorithm for job-shop scheduling,” *J. Softw.*, vol. 5, no. 3, pp. 269–274, 2010.
- [15] H. Nazif, “A Genetic Algorithm for Solving Scheduling Problem,” *J. Math. Comput. Sci.*, vol. 5, no. 2, pp. 91–96, 2012.
- [16] Z. Zuhri, *Algoritma Genetika, Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: ANDI Publisher, 2014.
- [17] Suyanto, *Evolutionary Computation (komputasi Berbasis Evolusi Dan Genetika)*. Bandung, Indonesia: Informatika, 2012.
- [18] E. K. Burke and S. Petrovic, “Recent Research Directions in Automated Timetabling,” *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, 2002.
- [19] N. K. Mawaddah and W. F. Mahmudy, “Optimasi Penjadwalan Perawat Menggunakan Algoritma Genetika,” *Univ. Brawijaya*, vol. 2, no. 2, pp. 1–8, 2006.