

BAB 2

LANDASAN TEORI

2.1 Penelitian Pendahuluan

Tabel 2.1 merupakan ringkasan dari penelitian-penelitian pendahuluan yang membahas mengenai Algoritma Genetika dalam menyelesaikan permasalahan penjadwalan.

Tabel 2.1. Penelitian pendahuluan

No	Penulis	Judul	Metode	Hasil
1	Tri Handayani, Dhomas Hatta Fudholi, dan Septia Rani	Kajian Algoritma Optimasi Penjadwalan Mata Kuliah [16]	Algoritma Greedy, Simulated Annealing, Algoritma Genetika	Algoritma Genetika dominan digunakan dalam menyelesaikan masalah penjadwalan dengan mampu mendapat solusi yang optimal dengan durasi waktu 0,964 hingga 73,461 detik dan fitness 1.
2	Imam Ahmad Ashari	Perbandingan Performansi Algoritma Genetika dan Algoritma Ant Colony Optimization Dalam Optimasi Penjadwalan Mata Kuliah [17]	Algoritma Genetika, Ant Colony Optimization	Waktu eksekusi Algoritma Genetika: 21,26 detik, waktu eksekusi Ant Colony Optimization: 69,11 detik. Memori digunakan Algoritma Genetika: 12.159,08 Kb, memori digunakan Ant Colony Optimization: 21.674,48 Kb.

Tabel 2.1. Penelitian pendahulu (Lanjutan)

No	Penulis	Judul	Metode	Hasil
3	Kartiko Agung Nugroho	Pembangunan Sistem Penjadwalan Petugas Ekaristi Berbasis Web Menggunakan Algoritma Genetika (Studi Kasus: Gereja Paroki Santa Maria Fatima Magelang) [21]	Algoritma Genetika	Sistem mengurangi waktu pembuatan jadwal secara <i>manual</i> dari yang awalnya 15 menit menjadi maksimal 5 menit.
4	Ahmed Redha Mahlous, dan Houssam Mahlous	Student Timetabling Genetic Algorithm Accounting for Student Preferences [3]	Algoritma Genetika	Algoritma Genetika terbukti dapat digunakan pada penjadwalan dengan menghasilkan solusi yang optimal ataupun mendekati optimal terhadap masalah yang kompleks. Pemilihan alokasi yang baik dapat menghasilkan hasil yang baik.

Tabel 2.1 menjelaskan penelitian lain yang membahas mengenai Algoritma Genetika dalam menyelesaikan masalah penjadwalan. Pada penelitian pertama di Tabel 2.1, penelitian tersebut membahas mengenai beberapa algoritma yang diterapkan pada permasalahan penjadwalan. Didapat hasil bahwa Algoritma Genetika yang digunakan menghasilkan hasil yang lebih baik dibanding algoritma lainnya, seperti Simulated Annealing dan Algoritma Greedy. Pada penelitian kedua, dilakukan perbandingan antara Algoritma Genetika dan Ant Colony Optimization

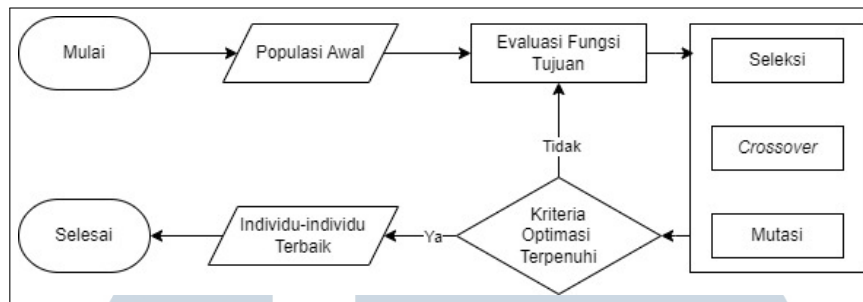
dan didapat hasil bahwa Algoritma Genetika lebih unggul dibanding Ant Colony Optimization dalam waktu eksekusi, yaitu 47,85 detik lebih cepat dan penggunaan memori 9.515,4 Kb lebih sedikit dibanding Ant Colony Optimization. Kemudian, pada penelitian ketiga dilakukan pembangunan sistem penjadwalan petugas ekaristi dengan Algoritma Genetika dan berhasil mempercepat hasil pembuatan jadwal petugas. Pada penelitian keempat, Algoritma Genetika diimplementasikan pada penjadwalan murid dan menghasilkan hasil yang optimal maupun mendekati optimal terhadap permasalahan yang kompleks. Dengan pemilihan alokasi yang baik, maka solusi yang dihasilkan oleh Algoritma Genetika juga akan menjadi baik.

2.2 Prodiakon

Kata prodiakon diambil dari bahasa Latin, yang terdiri dari *pro* dan *diakon*. Kata *pro* memiliki makna demi, untuk kepentingan, sebagai ganti, selaku, bagaikan, seolah-olah. Sementara itu, *diakon* diambil dari bahasa Yunani *diakonos* yang berarti seseorang yang melayani, membuat pelayanan, mengurus, dan menyelesaikan [12]. Prodiakon sendiri diangkat oleh keuskupan untuk melayani suatu tujuan dalam suatu wilayah tertentu. Pada umumnya, tugas prodiakon yang diberikan keuskupan adalah membangun dan memelihara persekutuan dengan orang lain, memberikan pelayanan kepada orang sakit, berbaur dengan kaum awam, membantu memimpin penyembahan, dan memimpin administrasi di gereja [22].

2.3 Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian yang memiliki basis pada mekanisme yang berdasar pada seleksi alam dan prinsip genetika. Algoritma ini cocok untuk menyelesaikan masalah optimasi yang kompleks [23]. Algoritma Genetika dimungkinkan untuk menjadi metode optimasi yang kuat di bidang komputasi evolusioner. Secara umum, cara kerja Algoritma Genetika digambarkan oleh Gambar 2.1.



Gambar 2.1. Bagan alur Algoritma Genetika [2]

2.3.1 Populasi Awal

Dalam Algoritma Genetika, langkah pertama yang dilakukan adalah melakukan inisialisasi terhadap populasi awal. Populasi awal adalah sekelompok individu yang menggambarkan sebuah solusi terhadap permasalahan yang ingin diselesaikan. Tiap individu akan direpresentasikan oleh sebuah kromosom dan pada tiap kromosom terdiri dari serangkaian gen. Masing-masing gen pada kromosom menggambarkan suatu karakteristik atau atribut dari solusi yang nantinya dihasilkan. Dalam Algoritma Genetika, salah satu jenis pengkodean atau *encoding scheme* yang diterapkan pada tiap kromosom atau gen adalah *value encoding*. Pada *value encoding*, kromosom dapat berupa *integer*, *real number*, karakter, atau objek [24]. Jenis pengkodean ini dapat digunakan untuk menyelesaikan masalah dengan nilai yang kompleks [25]. Populasi awal dibuat dengan menggunakan metode acak sehingga setiap kromosom pada populasi awal dapat mencakup berbagai kemungkinan solusi terhadap permasalahan penjadwalan. Ukuran populasi 50 hingga 100 memberikan hasil yang lebih baik pada Algoritma Genetika [3].

2.3.2 Evaluasi Fungsi

Evaluasi fungsi merupakan tahapan yang paling krusial dalam proses Algoritma Genetika untuk menentukan solusi terbaik [2]. Tahap ini akan dilakukan ketika populasi awal selesai dibangun dengan menguji kualitas kromosom menggunakan suatu fungsi *fitness*. Fungsi *fitness* ditunjukkan oleh Rumus 2.1

$$F(x) = \frac{1}{1 + \text{penalty}} \quad (2.1)$$

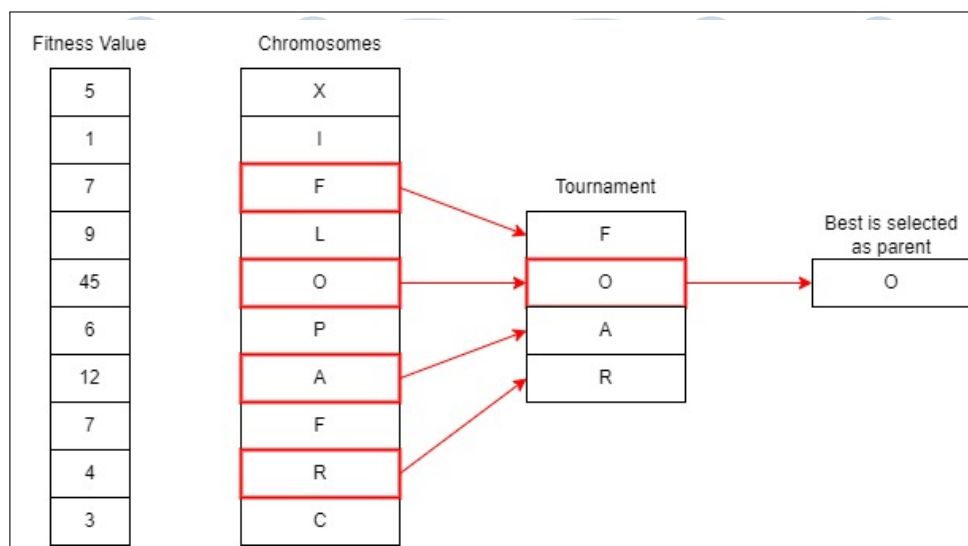
Pada fungsi fitness di Rumus 2.1, nilai *fitness* diperoleh dari nilai *penalty*

yang mempertimbangkan batasan terhadap masalah yang akan diselesaikan. Apabila solusi yang dihasilkan melanggar batasan, maka nilai *penalty* akan bertambah yang akan mempengaruhi nilai *fitness*. Semakin tinggi nilai *fitness*, semakin bagus solusi yang dihasilkan dan begitu pula sebaliknya.

2.3.3 Seleksi

Seleksi adalah tahapan penting pada Algoritma Genetika yang dapat dikatakan sebagai operator reproduksi. Proses ini akan menentukan apakah suatu individu layak untuk mengikuti proses reproduksi [25]. Pada umumnya, individu yang lebih baik cenderung untuk dipilih untuk dilanjutkan ke proses kawin silang. Salah satu jenis seleksi yang umum digunakan adalah *tournament selection* [3].

Tournament selection diperkenalkan pada tahun 1913 oleh Brindle [25]. Pada metode ini, sejumlah individu akan dipilih secara acak dan individu dengan *fitness* terbaik akan dipilih sebagai seorang *parent*. Hal ini juga akan dilakukan dalam pemilihan *parent* selanjutnya. Dalam permasalahan *timetabling*, *tournament selection* merupakan seleksi yang terbukti paling efektif karena lebih efektif dan dapat menangani *fitness* dengan nilai negatif. Ukuran *tournament* untuk mencegah adanya individu baik yang dominan adalah 5 [3]. Proses *tournament selection* ditunjukkan Gambar 2.2.



Gambar 2.2. *Tournament selection* [3]

2.3.4 Crossover

Tahapan *crossover* merupakan tahapan di mana beberapa gen pada dua buah kromosom induk atau *parent* saling ditukarkan antar satu sama lain sehingga terbentuk suatu kromosom baru yang mewarisi karakteristik dari induknya. [2]. Tahapan *crossover* ini bertujuan untuk membuat keturunan atau kromosom baru yang memiliki sifat dari dua induk berbeda, serta meningkatkan variasi pada populasi sehingga dapat membentuk solusi yang lebih optimal. Untuk mencegah individu dengan nilai *fitness* yang tinggi menjadi dominan, probabilitas *crossover* dibuat menjadi 0,9. Ada beberapa jenis *crossover*, yaitu [4]:

- *Single Point Crossover*

Pada jenis *crossover* ini, dua induk kromosom akan dipotong menjadi dua kepala dan dua ekor. Nantinya bagian-bagian tersebut akan ditukar secara acak, misalnya kedua ekor pada dua kromosom saling ditukar dan menciptakan individu baru [4]. Gambar 2.3 merupakan gambaran dari *single point crossover*.

Induk 1	1	2	3	4	5	6	7	8
Induk 2	9	10	11	12	13	14	15	16
Anak 1	1	2	3	4	5	14	15	16
Anak 2	9	10	11	12	13	6	7	8

Gambar 2.3. *Single point crossover* [4]

- *Multiple Point Crossover*

Pada jenis *crossover* ini, akan terjadi lebih dari satu titik potong dan terdapat ada banyak *crossover* atau segmen pada dua kromosom yang nantinya akan ditukar antar segmen secara acak. Gambar 2.4 merupakan gambaran dari *multiple point crossover*.

Induk 1	1	2	3	4	5	6	7	8
Induk 2	9	10	11	12	13	14	15	16
Anak 1	1	2	11	12	13	6	7	8
Anak 2	9	10	3	4	5	6	7	8

Gambar 2.4. *Multiple point crossover* [4]

- *Uniform Crossover*

Pada jenis *crossover* ini, gen yang dimiliki oleh anak akan didapat dari

menyalin gen dari induknya. Gen dari kedua induk akan dipilih secara acak dari indeks gen ke berapa dan dari induk mana. Gambar 2.5 merupakan gambaran dari *uniform crossover*.

Induk 1	1	2	3	4	5	6	7	8
Induk 2	9	10	11	12	13	14	15	16
Anak 1	1	2	9	10	5	6	12	14

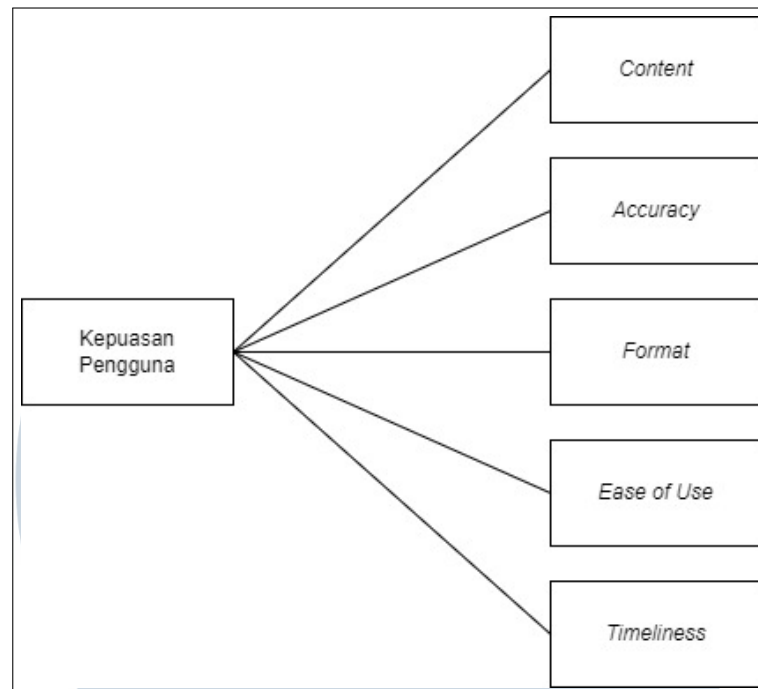
Gambar 2.5. *Uniform crossover* [4]

2.3.5 Mutasi

Pada tahapan Algoritma Genetika, mutasi adalah tahapan yang secara acak mengubah beberapa gen dalam kromosom. Hal ini bertujuan untuk membuat variasi genetik baru pada suatu kromosom sehingga kromosom tersebut memiliki peluang untuk memperluas ruang solusi [2]. Dengan adanya variasi genetik yang baru dibanding sebelumnya, Algoritma Genetika bisa menghindari kondisi yang sama. Probabilitas mutasi yang baik pada Algoritma Genetika berada pada angka 0,001 hingga 0,0001 [3]. Angka yang terlalu kecil dapat membuat nilai *fitness* terjebak dalam nilai optimum lokal, sedangkan angka yang terlalu besar dapat mempersulit mendapatkan nilai konvergen [26].

2.4 EUCS (End User Computing Satisfaction)

EUCS atau *End User Computing Satisfaction* merupakan model yang digunakan untuk mengukur tingkat kepuasan pengguna akhir dalam memanfaatkan suatu sistem informasi [27]. Model EUCS lebih berfokus pada kepuasan pengguna yang didasarkan pada pengalaman pengguna dalam menggunakan sistem informasi. Komponen yang terlibat dalam EUCS terdiri dari *content* (isi), *accuracy* (akurasi), *format* (tampilan), *ease of use* (kemudahan penggunaan), dan *timeliness* (waktu) [28].



Gambar 2.6. Model Evaluasi EUCS [1]

Berdasarkan Gambar 2.6, masing-masing komponen pada Model EUCS dijelaskan sebagai berikut:

- **Content**
Pada bagian *content*, kepuasan pengguna dinilai dari informasi yang disediakan sistem. Semakin lengkap modul dan kelengkapan sistem informasi, semakin tinggi kepuasan pengguna.
- **Accuracy**
Pada bagian *accuracy*, kepuasan pengguna dinilai dari keakuratan informasi yang dihasilkan oleh sistem ketika sistem menerima masukan.
- **Format**
Pada bagian *format*, kepuasan pengguna diukur melalui pengujian maupun evaluasi terhadap sistem tertentu, termasuk format data, informasi yang digunakan dalam sistem, serta faktor lain yang mempengaruhi kemampuan pengguna dalam menggunakan sistem.
- **Ease of Use**
Pada bagian *ease of use*, kepuasan pengguna diukur berdasarkan kemudahan

penggunaan sistem dan melakukan *input*, termasuk proses pengumpulan data, pengorganisasian data, serta analisis informasi yang diperoleh.

- Timeliness

Pada bagian *timeliness*, kepuasan pengguna dinilai dari kemampuan sistem dalam menyajikan informasi yang dibutuhkan oleh pengguna dalam waktu yang cepat.

Tabel 2.2 adalah tabel yang menjelaskan interval tingkat kepuasan pengguna.

Tabel 2.2. Interval persentase tingkat kepuasan [1]

Interval	Kriteria
0% - 20%	Sangat Tidak Puas
21% - 40%	Tidak Puas
41% - 60%	Cukup
61% - 80%	Puas
81% - 100%	Sangat Puas

Untuk mengetahui kategori responden, terdapat satu teknik analisis yang dapat dilakukan, yaitu Metode Likert. Langkah-langkah pada Metode Likert dapat dilihat pada 2.2 hingga 2.4.

$$\sum SK = ST \times JP \times JS \quad (2.2)$$

Keterangan Rumus 2.2:

- SK adalah skor kriterium (skor ideal)
- ST adalah skor paling tinggi tiap pertanyaan
- JP adalah jumlah pertanyaan
- JS adalah jumlah responden

$$SH = T \times Pn \quad (2.3)$$

Keterangan Rumus 2.3:

- T adalah banyaknya responden
- Pn adalah skor likert yang dipilih

$$P = \frac{\sum SH}{\sum SK} \times 100\% \quad (2.4)$$

Rumus 2.4 adalah rumus untuk menentukan hasil akhir dari jawaban responden dan dinyatakan dalam bentuk persen. Hasil yang didapat nantinya akan dikategorikan berdasarkan Tabel 2.2.

2.5 Skala Likert

Skala Likert diciptakan oleh Rensis Likert, yang menciptakan sebuah skala untuk mengukur suatu sikap. Pengukuran pada Skala Likert dilakukan dengan menanyakan kepada orang-orang seberapa setuju atau tidak setuju orang tersebut terhadap pernyataan di topik tertentu [29]. Skala Likert dengan 5 tingkat merupakan jenis yang paling sering digunakan dalam penelitian karena memudahkan responden dalam memilih jawaban. Skala ini terdiri dari lima pilihan, yaitu Sangat Setuju, Setuju, Netral, Tidak Setuju, dan Sangat Tidak Setuju. Skala Likert memiliki 2 bentuk pertanyaan, yaitu positif dan negatif. Pertanyaan positif digunakan untuk mengukur skala positif dengan skor 5, 4, 3, 2, 1. Pertanyaan negatif digunakan untuk mengukur skala negatif dengan skor 1, 2, 3, 4, 5 [30].

