



Implementasi Metode Algoritma Genetika untuk Penjadwalan Mata Kuliah pada Universitas Dinamika Bangsa

Agus Nugroho¹

¹ Ilmu Komputer, Teknik Informatika, Universitas Dinamika Bangsa, Jl. Jendral Sudirman, Thehok, Jambi 36138, Indonesia.

ABSTRACT

The activity of preparing the class schedule is a routine for the academic side of the Academic and Student Administration Bureau (BAAK) every semester, the process of compiling a class schedule takes quite a long time due to considerations that must be regulated, one of which is a request from the lecturer due to the availability of time to teach certain hours. On the problems that have been presented, the authors want to help optimize and facilitate activities in the preparation of class schedules by considering the existing rules. Algorithm is a method that can optimize the value through the search process. The optimal value is obtained from population generation, in which each population will be selected and cross over to the best population, so that the evolution of a new population appears based on the value of the mutation rate. The human method can be used to optimize the course scheduling process at Dinamika Bangsa University, so that the work process can be assisted to work faster. The result is a population that has achieved an overall fitness value of 1.0 as a condition for stopping the process which is fulfilled in the 4,270,701 generation with a time of 09:00:39, to produce a schedule of 484 people with 10 populations. Implementation of algorithmic methods that can be used to optimize and facilitate lecture activities with the rules of the availability of lecturer time.

Keywords: Genetic Algorithm; Scheduling; Cross Over; Algorithm; Implementation

ABSTRAK

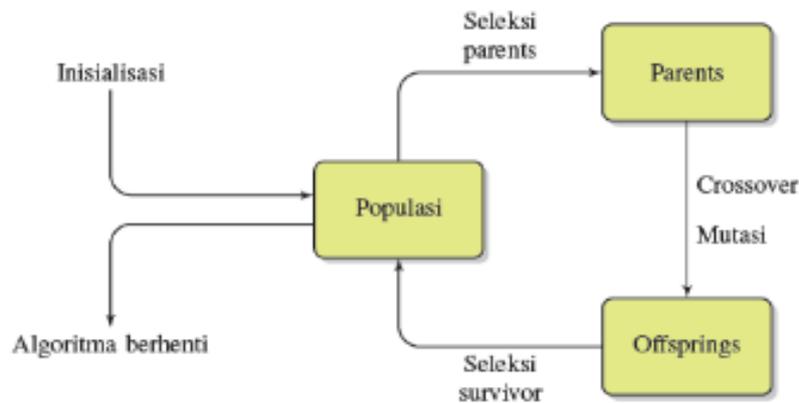
Kegiatan penyusunan jadwal kuliah merupakan rutinitas dari pihak akademik bagian Biro Administrasi Akademik dan Kemahasiswaan (BAAK) tiap semesternya, proses penyusunan jadwal mata kuliah memakan waktu yang cukup lama dikarenakan banyaknya pertimbangan yang harus diatur, salah satunya yaitu permintaan dari dosen karena ketersediaan waktu untuk dapat mengajar pada jam tertentu. Pada permasalahan yang telah disampaikan, penulis ingin mencoba membantu mengoptimalkan dan mempermudah kegiatan dalam penyusunan jadwal mata kuliah dengan pertimbangan rule-rule yang ada. Algoritma genetika merupakan metode yang dapat mengoptimalkan sebuah nilai melalui proses pencarian. Nilai yang optimal didapat dari generasi populasi, yang mana tiap populasi akan dilalukan seleksi serta dilakukan cross over terhadap populasi terbaik, sehingga muncul evolusi dari populasi yang baru berdasarkan besarnya nilai dari laju mutasi. Metode algoritma genetika dapat dimanfaatkan untuk mengoptimalkan proses penjadwalan mata kuliah pada Universitas Dinamika Bangsa, sehingga proses kerja manusia dapat dibantu untuk berkerja lebih cepat. Hasilnya populasi yang telah mencapai nilai fitness keseluruhan 1.0 sebagai syarat pemberhentian proses yang dipenuhi pada generasi ke 4.270.701 dengan lama waktu 09:00:39, untuk mengenerate jadwal sebanyak 484 gen dosen dengan 10 populasi. Implementasi metode algoritma genetika dapat digunakan untuk mengoptimalkan dan mempermudah penyusunan jadwal perkuliahan dengan aturan/ rule dari ketersediaan waktu dosen.

Kata Kunci: Algoritma Genetika; Penjadwalan; Crossover; Algoritma; Implementasi

1. PENDAHULUAN

Berbicara mengenai perkembangan teknologi belakangan mengalami kemajuan yang sangat pesat sekali, terutama dalam penyelesaian masalah-masalah yang terjadi dalam kehidupan manusia. Temuan pada metode-metode tertentu yang di implementasikan merupakan sebuah kemajuan pada kehidupan manusia dalam memanfaatkan teknologi, salah satunya adalah metode algoritma genetika. Dalam kegiatan penyusunan jadwal kuliah merupakan rutinitas dari pihak akademik bagian Biro Administrasi Akademik dan Kemahasiswaan (BAAK) tiap semesternya, proses penyusunan jadwal mata kuliah memakan waktu yang cukup lama dikarenakan banyaknya pertimbangan dalam penyusunan. Salah satu pertimbangan yang harus diatur adalah permintaan dari dosen karena ketersediaan waktu untuk dapat mengajar pada jam tertentu. Hal tersebut menyebabkan penyusunan jadwal matakuliah harus disesuaikan dengan ketersediaan dosen mengajar pada mata kuliah tertentu. Pada kasus yang telah disampaikan penulis ingin mencoba mengimplementasikan metode Algoritma Genetika yang dapat membantu mengoptimalkan dan mempermudah kegiatan dalam penyusunan jadwal mata kuliah dengan banyaknya pertimbangan rule-rule yang ada. [1] Tahapan-tahapan dari Algoritma genetika antara lain adalah diawali dengan Pembentukan Kromosom dari Populasi yang terdiri dari beberapa gen. Suatu kromosom digunakan untuk merepresentasikan suatu permasalahan. Kromosom bisa dinyatakan dalam beberapa cara, antara lain seperti kromosom biner, kromosom real dan kromosom permutasi serta lain sebagainya. Kumpulan suatu kromosom ddalam algoritma genetika dinamakan dengan populasi. Dalam satu populasi, akan terdapat N buah kromosom dengan nilai N adalah suatu parameter yang ditetapkan oleh user. Fungsi fitness memberikan nilai dengan tingkat kecocokan pada sebuah kromosom yang ada pada suatu populasi. Semakin tinggi nilai fitness, seharusnya solusi tersebut semakin optimal. Proses Seleksi Parents, Proses ini bertujuan untuk memilih dua kromosom sebagai parent yang selanjutnya akan dilakukan proses crossover dan mutasi pada parent terpilih. Proses Crossover dan Mutasi adalah proses persilangan, yaitu membentuk dua offspring (kromosom anak) baru dari dua parent. Hanya offspring yang memenuhi syarat pada permasalahan yang akan ditambahkan ke populasi yang ada. Proses mutasi yaitu mengganti suatu gen dengan gen yang baru. Proses crossover dan mutasi biasanya dilakukan secara acak, yang mana pada penelitian lainnya yang dilakukan Nasution [2] menganalisis solusi dari TSP penggunaan *cross over partially mapped* yang bisa menentukan

nilai dari probabilitas *crossover* dari 20%, 40%, 60%, 80%, serta 90%. Pada penelitian lainnya yang dilakukan Kusum Deep & Hadush Mebrahtu [3], *crossover* yang dilakukan dilakukan dengan variasi yang mana kromosom diletakan secara acak. Al kasasbeh [4] menambahkan sebuah prosedur baru untuk menyelesaikan permasalahan TSP dengan cara menambahkan metode *shared neighbor*. Proses Seleksi Survivor atau Replacement bertujuan untuk memilih N kromosom dari gabungan antara populasi sebelumnya dan offspring yang dihasilkan. Lalu, N kromosom yang terpilih akan digunakan sebagai populasi untuk perulangan atau iterasi berikutnya. Syarat Pemberhentian jika ada kromosom atau solusi yang optimal secara global, nilai fitness-nya telah mencapai konvergensi, atau iterasinya telah mencapai jumlah maksimum yang ditentukan oleh user. [5]



Gambar 1. Proses Algoritma Genetika [1]

Dari penelitian sebelumnya yang pernah dilakukan dengan judul “Optimalisasi Rute Obyek Wisata Di Bandung Raya Menggunakan Algoritma Genetika”. Mendapatkan kesimpulan bahwa Hasil akhir dari penelitian ini adalah memberikan rekomendasi rute obyek wisata terdekat sebanyak lima tujuan yang dipilih oleh pengguna serta membandingkan lima tujuan rute obyek wisata untuk mencari jarak terpendek dan waktu tercepat, [6]. Pada penelitian yang lainnya yang dilakukan dengan judul “Implementasi Algoritma Genetika untuk Memprediksi Waktu dan Biaya Pengerjaan Proyek Konstruksi”, [7]. Mendapatkan kesimpulan yaitu Aplikasi ini dapat memberikan output berupa hasil prediksi waktu dan biaya pengerjaan proyek konstruksi berdasarkan luas tanah yang di input dan pemilihan kromosom. Kromosom yang dimaksud adalah salah satu pekerjaan dari masing-masing proyek, yang memiliki luas tanah dengan selisih 5 angka dari luas tanah yang di input. Algoritma Genetika dapat diimplementasi pada sistem. Hal ini dibuktikan berdasarkan hasil output berupa hasil prediksi biaya dan waktu pengerjaan proyek menggunakan Algoritma Genetika dengan persentase keakuratan sebesar 98.72%. Pada Penelitian yang dilakukan oleh servita dkk [8], dengan judul penelitian Optimalisasi Program Kerja Organisasi Mahasiswa Di Unjani Menggunakan Algoritma Genetika didapat kesimpulan bahwa dapat menghasilkan jadwal kerja dengan sangat optimal, sehingga menjadi solusi yang dapat digunakan untuk mencegah terjadinya kemunduran program kerja. Penelitian yang dilakukan oleh Purwanto dkk [9], berjudul Optimalisasi Penempatan Halte Trans Metro Bandung Menggunakan Algoritma Genetika, dapat juga digunakan sebagai penempatan objek untuk mendapatkan jarak yang optimal, yang mana dalam penelitiannya disimpulkan bahwa terdapat beberapa kelemahan dari system yang mana suatu individu dengan solusi yang kurang baik akan bertahan selama beberapa generasi. Santika Dewi [10], dengan penelitiannya berjudul Optimalisasi Penempatan Guru Sekolah Dasar di Kecamatan Cikajang Kabupaten Garut Menggunakan Algoritma Genetika, berkesimpulan bahwa penempatan para guru bisa dilakukan berdasarkan mempertimbangkan beberapa attribute, diantaranya kualifikasi, usia, pendidikan, pengalaman mengajar, status dan kebutuhan guru pada sekolah. Optimalisasi lahan tanah untuk area rumah dan jalan menggunakan algoritma genetika ditulis oleh Luthfi Ahmad Fadhil dkk [11], mendapatkan kesimpulan bahwa system ini dapat membuat sketsa terbentuknya denah untuk penempatan area rumah serta jalan dalam bentuk kaveling-kaveling.

Dari penelitian sebelumnya yang pernah dengan judul “Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus Pada Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako)” [5]. Proses penjadwalan diawali dengan proses pembangkitan cromossom, setelah itu populasi dibentuk, dilanjutkan dengan penetapan nilai fitness, barulah masuk ke tahap proses seleksi. Selanjutnya kromosom yang telah melalui tahap seleksi dibangkitkan dengan nilai *random*. hasil yang didapat dilakukan proses *crossover* dan proses mutasi. langkah terakhir dilakukan penggabungan kromosom *offspring*, hasil dari proses mutasi dan kromosom seleksi sebelumnya. Penjadwalan dilakukan dengan menginputkan 25 dosen dan 45 mata kuliah serta 10 ruang disemester genap pada tahun ajaran 2016/2017. Hasil dari fitnes terbaik diketemukan pada generasi ke 2127 dengan lama waktu komputasi sekitar 6 jam 36 menit. Proses algoritma berjalan dengan optimal karena tidak terdapat jadwal yang konflik.

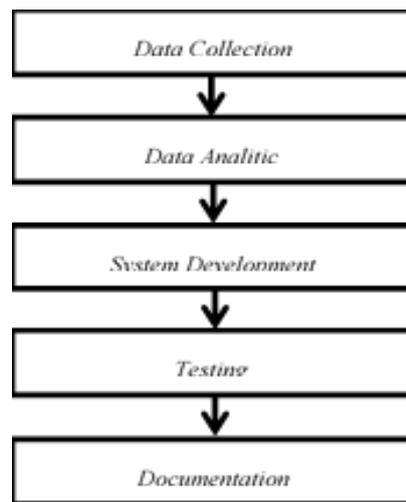
Dari latar belakang permasalahan yang terpapar diatas penulis mencoba mengimplementasikan metode algoritma genetika untuk penjadwalan matakuliah pada universitas dinamika bangsa, sehingga permasalahan yang dihadapi pihak BAAK dapat diselesaikan. Berdasarkan uraian pada latar belakang diatas, maka dapat disimpulkan permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana mengimplementasikan metode Algoritma Genetika untuk penjadwalan matakuliah menggunakan pada Universitas Dinamika Bangsa

2. METODE PENELITIAN

2.1. Alur Kerangka Kerja Penelitian (*Research Framework*)

Suatu usaha untuk menemukan biasa disebut dengan penelitian, pengembangan, dan melakukan pembuktian untuk mendapatkan kebenaran berdasarkan pengujian yang dilakukan dengan berbagai metode yang bersifat ilmiah. Juga dikatakan sebagai alur untuk mendapatkan pemecahan masalah secara ilmiah. Dalam sebuah penelitian, penulis menggunakan langkah-langkah yang terperinci untuk mendapatkan hasil penelitian yang bermanfaat serta dapat dipertanggung jawabkan.

Dalam memulai suatu penelitian, tahapan-tahapan dalam pelaksanaannya harus tergambar diawal agar kepedan mendapatkan penelitian dengan hasil berdasarkan metode ilmiah yang telah diukur berdasarkan pengujian. Kerangka kerja penelitian (*research framework*) adalah sebuah bentuk berdasarkan kepada tahap-tahap yang disusun dengan *systematic* yang kemudian bisa diimplementasikan sebagai langkah penyelesaian penelitian. Langkah-langkah yang dilakukan peneliti sebagai berikut:



Gambar 2. Research Framework

Dari gambar 2 diatas, penguraian serta pembahasan masing-masing langkah penelitian ini adalah sebagai berikut:

1. *Data Collection*
Langkah pada proses ini, melakukan pengumpulan data dengan cara penulis melakukan koleksi berupa data-data pustaka yang ada kaitannya dengan dengan Metode Algoritma Genetika.
2. *Data Analytic*
Pada proses ini, penulis melakukan analisa data-data yang telah didapatkan pada tahapan sebelumnya.
3. *System Development*
Tahap ini dilakukan analisa kebutuhan terkait dengan penjadwalkan matakuliah dengan nilai-nilai tertentu menggunakan metode Algoritma Genetika. Proses dilanjutkan dengan membuat model flow untuk mengetahui alur dari aplikasi yang akan dikembangkan.
4. *Testing*
Proses pegujian aplikasi penjadwalan mata kuliah yang telah dikembangkan dengan penerapan metode algoritma genetika menggunakan metode *blackbox*.
5. *Documentation*
Proses ini adalah melakukan dokumentasi dalam bentuk laporan agar setiap langkah penelitian dapat ditulis dan dibuktikan secara ilmiah.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Metode Algoritma Genetika

Setelah melakukan analisis terhadap system yang berjalan, penulis mencoba melakukan implementasi metode algoritma genetika untuk membuat penjadwalan berjalan secara otomatis berdasarkan mutasi populasi sehingga mendapatkan keluaran yang sesuai.

3.1.1 Inisialisasi

Sebelum menjadikan gen masuk kedalam populasi, yang pertama dilakukan yaitu menginisialisasikan model class dosen yang akan digunakan masuk sebagai gen awal. Adapun properties/ field yang dibutuhkan untuk model class dosen adalah 19 field, seperti terlihat pada potongan code dibawah.

```
class Dosen
{
    public string nama;
    public string mata_kuliah;
    public string kode_mata_kuliah;
    public int sks;
    public int kelas_pagi;
    public int kelas_malam;
    public string kode_ruang;
    public string[] waktu;
    public string hari;
    public int index_hari;
    public string ruang;
    public string kecuali_hari;
    public string hanya_hari;
    public bool kesesuaian_kecuali_hari;
    public bool kesesuaian_hanya_hari;
    public bool status;
    public bool kesesuaian_waktu;
    public bool kesesuaian_ruang;
    public bool kesesuaian_jam;
}
```

Gambar 3. Propertis/Field pada class dosen

Class dosen diatas dimasukan kedalam variable List<Dosen> sebanyak ...n, dimana akan dilakukan random pada tahap selanjutnya. List<Dosen> yang dimasukan akan dijadikan gen, yang mana gen ini akan dirandom sebanyak Populasi yang diperlukan.

3.1.2 Populasi

Populasi merupakan kumpulan dari gen yang sudah diinisialisasi berdasarkan data awal dosen, pada sebuah populasi gen dimasukan secara acak. Jumlah populasi mulai dari 3 sampai ...n. yang mana masing-masing populasi ini akan dinilai dan diambil 2 populasi terbaik. Berikut potongan code berdasarkan nilai acak yang diambil dari variable List<Dosen> untuk dijadikan gen pada sebuah populasi.

```
for (int populasi = 0; populasi < populasi.Count; populasi++){
    for (int doseni = 0; doseni < total_dosen; doseni++){
        var ran_dosen = random.Next(0, dosens.Count);
        var d = dosens[ran_dosen];
        populasi[populasi].dosen.Add(new Dosen(
            d.nama, d.kode_mata_kuliah,
            d.mata_kuliah,
            d.sks,
            d.kelas_pagi,
            d.kelas_malam,
            d.kode_ruang) {
                hanya_hari = d.hanya_hari,
                kecuali_hari = d.kecuali_hari });
        var m = populasi[populasi].dosen[doseni];
        int ran_hari = random.Next(0, ContentValues.hari.Length);
        m.hari = ContentValues.hari[ran_hari];
        m.index_hari = ran_hari;
        int ran_waktu = 0;
        if (m.sks == 3 || m.sks == 4) ran_waktu = random.Next(0, 6);
        if (m.sks == 2) ran_waktu = random.Next(7, ContentValues.waktu.Count);
        ContentValues.waktu[ran_waktu].CopyTo(m.waktu = new
            string[ContentValue.waktu[ran_waktu].Length], 0);
        int ran_ruang = 0;
        if (m.kode_ruang == "T") ran_ruang = random.Next(0, 7);
        if (m.kode_ruang == "J") ran_ruang = 7;
        if (m.kode_ruang == "L") ran_ruang = random.Next(8, 10);
        if (m.kode_ruang == "R") ran_ruang = random.Next(10, 26);
        m.ruang = ContentValues.ruang[ran_ruang];
    }
}
```

Gambar 4. Random gen masuk kedalam populasi

Dari potongan code diatas terdapat iterasi sebanyak dua kali. Iterasi pertama digunakan untuk iterasi banyaknya populasi dimulai dari populasi awal hingga ke populasi ...n. Iterasi kedua digunakan banyaknya jumlah dosen yang diacak masuk kedalam populasi mulai dari jumlah awal dosen hingga ke dosen ..n.

3.1.3 Seleksi Parent

Setelah tahap pembentukan Populasi sebanyak ...n, maka dilanjutkan pada bagian seleksi untuk mendapatkan 2 populasi dengan nilai fitness terbaik. 2 nilai populasi dengan nilai fitness terbaik diambil untuk dijadikan parent populasi.

```
for (int i = 0; i < 2; i++){
    var index = populasi.FindIndex(x => x.fitness_populasi.Equals(populasi.Max(y =>
        y.fitness_populasi)));

    var d = populasi[index].dosen;

    terbaik.Add(new Populasi(new List<Dosen>(d)) { fitness_populasi =
        populasi[index].fitness_populasi });

    mutant.Add(new Populasi(new List<Dosen>(d)) { fitness_populasi = populasi[index].fitness_populasi });
    populasi.RemoveAt(index);
}
```

Gambar 5. Iterasi sebanyak 2 kali untuk seleksi populasi

3.1.4 Crossover

Pada tahap Crossover, adalah menggabungkan setengah dari masing-masing populasi terbaik dengan cara dipersilangkan. Ada beberapa metode crossover, dua diantaranya penulis coba implementasikan yaitu metode persilangan satu titik (single-point crossover) dan persilangan banyak titik (*multi-point crossover*) sebagai perbandingan diakhir untuk menemukan metode crossover yang paling cocok digunakan dalam proses mutasi penjadwalan.

```
for (int i = 0; i < mutant[0].dosen.Count / 2; i++){  
    mutant[0].dosen[i] = terbaik[1].dosen[i];  
    mutant[1].dosen[i] = terbaik[0].dosen[i];  
}
```

Gambar 6. Persilangan satu titik

```
for (int i = 0; i < mutant[0].dosen.Count; i+=2){  
    mutant[0].dosen[i] = terbaik[1].dosen[i];  
    mutant[1].dosen[i] = terbaik[0].dosen[i];  
}
```

Gambar 7. Persilangan banyak titik Mutasi

Pada tahap ini parent populasi dilakukan proses mutasi, yaitu melakukan proses nilai acak float antara 0.0 hingga 1.0 tergantung dari besaran laju mutasi (rate mutation), yang mana dilakukan iterasi sebanyak gen yang ada pada masing-masing parent populasi, setiap gen pada populasi akan melalui iterasi dengan kondisi blocking jika nilai acak berada dibawah laju mutasi, maka gen tersebut akan dimasukkan ulang dengan nilai baru tentunya dengan melakukan pengacakan dari sumber gen.

3.1.5 Mengembalikan populasi hasil mutasi

Setelah melakukan mutasi, populasi baru akan dihitung ulang fitness populasinya untuk menggantikan dua populasi dengan nilai fitness terburuk pada populasi asal. Adapun pada penelitian ini penulis membuat perhitungan nilai fitness keseluruhan yang terdiri dari 6 nilai fitness. Adapun diantaranya :

1. Status
2. Kesesuaian_Jadwal
3. Kesesuaian_Jam
4. Kesesuaian_Ruang
5. Kesesuaian_Hanya_Hari
6. Kesesuaian_Kecuali_hari

Dari ke enam nilai fitness tersebut digabungkan dan dibagi sebanyak parameter fitness, sehingga didapat total fitness. Adapun function dalam bentuk code sebagai berikut.

```
public void HitungFitnessPopulasi(  
    float status,  
    float kesesuaian_jadwal,  
    float kesesuaian_jam,  
    float kesesuaian_ruang,  
    float kesesuaian_hanya_hari,  
    float kesesuaian_kecuali_hari,  
    float total_dosen  
)  
{  
    fitness_populasi =  
    ((kesesuaian_jadwal + status + kesesuaian_jam +  
    kesesuaian_ruang +  
    kesesuaian_hanya_hari+kesesuaian_kecuali_hari)  
    / total_dosen) / 6;  
}
```

Gambar 8. Perhitungan fitness untuk tiap class populasi

3.1.6 Syarat Pemberhentian

Syarat pemberhentian dilakukan untuk menghentikan proses pengulangan populasi yang selalu melalui tahapan-tahapan hingga menghasilkan populasi baru yang sudah bermutasi. Jika ada nilai populasi yang telah memenuhi syarat pemberhentian, maka proses pengulangan akan dihentikan. Berikut syarat yang dilakukan penulis dalam bentuk code, yang mana nilai fitness dari salah satu banyaknya populasi harus bernilai 1.0.

```

foreach (var item in populasi){
    if (item.fitness_populasi >= 1f){
        ketemu = true;
        Console.Beep(400, 200);
        break;
    }
}
}

```

Gambar 9. Penghentian iterasi pengecekan nilai fitness

Jika pada pengecekan semua populasi belum ada yang memenuhi syarat penghentian, maka proses pengulangan akan dikembalikan ke proses seleksi populasi kembali, hingga syarat penghentian terpenuhi.

4. HASIL IMPLEMENTASI METODE ALGORITMA DAN GENETIKA

Pada penelitian ini penulis melakukan inisialisasi awal banyak populasi adalah 10 dengan nilai laju mutasi sebesar 0.1, dengan banyaknya gen adalah 484 yang mana diambil dari jadwal dosen program studi Teknik Informatika dan Sistem Informasi saja. Pada gambar dibawah terlihat ada 10 populasi pada generasi pertama dengan nilai fitness masing-masing, yang mana 2 diantaranya adalah populasi dengan nilai fitness terbaik, dan 2 populasi terakhir adalah populasi hasil mutasi.

```

C:\Windows\system32\cmd.exe
generasi : 1
populasi
0.5404762
0.5297619
0.5702381
0.5738095
0.5690476
0.5583333
0.5738095
0.5738095
-----
terbaik
0.5797619
0.5785714
-----
mutant
0.577381
0.5583333

```

Gambar 10. Proses penjadwalan pada generasi 1

Rata-rata nilai fitness pada masing-masing populasi berkisar sekitar 0.5.

```

C:\Windows\system32\cmd.exe
generasi : 10000
populasi
0.8452381
0.8440476
0.8440476
0.8297619
0.8440476
0.8428571
0.8428571
0.8464286
-----
terbaik
0.8547619
0.8464286
-----
mutant
0.8202381
0.8238095

```

Gambar 11. Proses penjadwalan pada generasi 10.000

Dari gambar diatas terlihat beberapa populasi sudah mencapai nilai fitness sekitar 0.84 dan tidak banyak perubahan signifikan pada generasi ke 10.000.

```

C:\Windows\system32\cmd.exe
Rabu 4.6 12.30-13.15 13.15-14.00 14.00-14.45 Sistem Digital - Marrylinteri Istoningtyas, ST, M.Kom status: True waktu: True
Rabu 2.10 14.45-15.30 15.30-16.15 16.15-17.00 Sistem Digital - Marrylinteri Istoningtyas, ST, M.Kom status: True waktu: True
Senin 4.2 12.30-13.15 13.15-14.00 14.00-14.45 Sistem Digital - Marrylinteri Istoningtyas, ST, M.Kom status: True waktu: True
Sabtu 4.4 12.30-13.15 13.15-14.00 14.00-14.45 Sistem Digital - Marrylinteri Istoningtyas, ST, M.Kom status: True waktu: True
Rabu L.4.1 17.00-17.45 17.45-18.30 18.30-19.15 Pemrograman Berorientasi Objek - Martono, M.Kom status: True waktu: True
Sabtu L.4.1 19.15-20.00 20.00-20.45 20.45-21.30 Algoritma dan Struktur Data I - Martono, M.Kom status: True waktu: True
Rabu L.87 10.15-11.00 11.00-11.45 11.45-12.30 Pemrograman Berorientasi Objek - Roby Setiawan, S.Kom, M.S.I status: True waktu: True
Selasa 4.4 12.30-13.15 13.15-14.00 Bahasa Inggris II - Saiful Effendi, M.Pd status: True waktu: True
Selasa 3.7 11.00-11.45 11.45-12.30 Bahasa Inggris II - Saiful Effendi, M.Pd status: True waktu: True
Rabu 3.7 12.30-13.15 13.15-14.00 Bahasa Inggris II - Saiful Effendi, M.Pd status: True waktu: True
Jumat 4.5 12.30-13.15 13.15-14.00 Bahasa Inggris II - Saiful Effendi, M.Pd status: True waktu: True
Rabu 1.6 15.30-16.15 16.15-17.00 Bahasa Inggris II - Silvia Rianti Agustini, S.Pd, M.S.I status: True waktu: True
Jumat 3.7 18.30-19.15 19.15-20.00 Bahasa Inggris II - Silvia Rianti Agustini, S.Pd, M.S.I status: True waktu: True
Senin 4.7 15.30-16.15 16.15-17.00 Kalkulus II - Syamsyida Rozi, S.Si status: True waktu: True
Jumat 3.5 15.30-16.15 16.15-17.00 Kalkulus II - Syamsyida Rozi, S.Si status: True waktu: True
Sabtu 3.9 12.30-13.15 13.15-14.00 Kalkulus II - Syamsyida Rozi, S.Si status: True waktu: True
Selasa 4.5 15.30-16.15 16.15-17.00 Kalkulus II - Veny Sri Astuti, M.Pd status: True waktu: True
Rabu 4.8 15.30-16.15 16.15-17.00 Kalkulus II - Veny Sri Astuti, M.Pd status: True waktu: True
Sabtu 3.8 17.00-17.45 17.45-18.30 18.30-19.15 Sistem Digital - Yudi Novianto, S.Kom, M.S.I status: True waktu: True

-----
nilai fitness :1
ditemukan pada generasi ke :4270701
dengan lama waktu : 09:00:39.3384151
total dosen di generate : 484
Press any key to continue . . .

```

Gambar 12. Proses penjadwalan dengan nilai fitness 1.0

Dari gambar diatas terlihat merupakan gambar populasi yang telah mencapai nilai fitness keseluruhan 1.0, serta telah memenuhi syarat pemberhentian proses. Adapun ringkasan pada proses ini terlihat pada gambar dibawah.

```

-----
nilai fitness :1
ditemukan pada generasi ke :4270701
dengan lama waktu : 09:00:39.3384151
total dosen di generate : 484
Press any key to continue . . .

```

Gambar 13. Detil dari proses penjadwalan

Dari gambar diatas terlihat bahwa, nilai fitness 1.0 ditemukan pada generasi 4.270.701 dengan lama waktu 09:00:39, untuk mendapatkan jadwal 484 gen dosen dengan banyak populasi 10 populasi. Setiap pengulangan tercipta gen sebesar 484 dikali dengan 10 populasi, maka menghasilkan 4.840 gen setiap pengulangannya.

5. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan sebuah aplikasi penjadwalan yang mengimplementasikan metode algoritma genetika. Metode algoritma genetika digunakan untuk mengoptimalkan dan mempermudah pengguna dalam menyusun jadwal perkuliahan dengan aturan/ rule dari ketersediaan waktu dosen. Besaran populasi dalam melakukan random gen, mempengaruhi kecepatan dalam melakukan proses mutasi. Metode cross over tidak begitu terlalu berpengaruh besar dalam proses mutasi, melainkan nilai dari laju mutasi sangat berperan penting seberapa banyak nilai gen yang akan dirandom ulang. Semakin kecil nilai laju mutasi, maka akan semakin akurat, tetapi proses mutasi semakin sedikit.

DAFTAR PUSTAKA

- [1] A. Kholik, E. E. Wahyudi, K. Devianto, and N. Sholihah, "Sistem Rekomendasi Berbasis Genetic Algorithm : Studi Kasus Pembelian Komponen Komputer dan Aksesorisnya," *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2018*, pp. 30–34, 2018.
- [2] K. Nasution, "Analisis Pemilihan Partially Mapped Crossover Algoritma Genetika Pada Penyelesaian Travelling Salesman Problem," *Universitas Sumatera Utara Medan*, 2011.
- [3] K. Deep, "New Variations of Order Crossover for Travelling Salesman Problem Workshop on 'Stochastic Modeling and Optimal Control of Engineering Systems' View project Coordination of Directional Over current Relays View project Redalyc Scientific Information System," 2011. [Online]. Available: <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=265219618002>
- [4] A. A. T. A.-R. Mouhammd Al kasassbeh, *Shared Crossover Method for Solving Traveling Salesman Problem*, vol. IIJCS Volume 1. Canadian Intellectual Property Office, 2012. Accessed: Mar. 08, 2022. [Online]. Available: www.ij-cs.org
- [5] A. T. Saud, D. W. Nugraha, and A. Y. E. Dodu, "Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus Pada Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako)," *Jurnal Ilmiah Matematika Dan Terapan*, vol. 14, no. 2, pp. 242–255, 2017, doi: 10.22487/2540766x.2017.v14.i2.9026.
- [6] N. M. Hasyim, E. C. Djamal, and A. Komarudin, "Optimalisasi Rute Obyek Wisata Di Bandung Raya Menggunakan Algoritma Genetika," *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, pp. 6–10, 2017, [Online]. Available: file:///D:/Kuliah UAJY/Semester 7/PPTA/GA_Wisata2.pdf

- [7] K. Krisnandi and H. Agung, "Implementasi Algoritma Genetika Untuk Memprediksi Waktu Dan Biaya Pengerjaan Proyek Konstruksi," *Jurnal Ilmiah FIFO*, vol. 9, no. 2, p. 90, 2017, doi: 10.22441/fifo.2017.v9i2.001.
- [8] B. Servitia, E. C. Djamal, and U. Jenderal Achmad Yani Jl Terusan Jenderal Sudirman, "Optimalisasi Program Kerja Organisasi Mahasiswa Di Unjani Menggunakan Algoritma Genetika."
- [9] F. Purwanto, E. C. Djamal, A. Komarudin, and U. Jenderal Achmad Yani Jl Terusan Sudirman, "Optimalisasi Penempatan Halte Trans Metro Bandung Menggunakan Algoritma Genetika," 2016.
- [10] O. Penempatan Guru *et al.*, *Prosiding Seminar Nasional Komputer dan Informatika (SENASKI)*.
- [11] L. Ahmad Fadhil, E. C. Djamal, R. Ilyas Jurusan Informatika, F. Mipa, and U. Jenderal Achmad Yani Jl Terusan Jenderal Sudirman, *Optimalisasi Lahan Tanah untuk Area Rumah dan Jalan ... (Fadhil dkk.) OPTIMALISASI LAHAN TANAH UNTUK AREA RUMAH DAN JALAN MENGGUNAKAN ALGORITMA GENETIKA*.