

PEMODELAN SISTEM OPTIMASI PENJADWALAN MATAKULIAH DENGAN ALGORITMA GENETIKA

(COURSE SCHEDULING OPTIMIZATION SYSTEM MODELING WITH GENETIC ALGORITHMS)

Eka Yulia Sari¹⁾, Dina Yulina Heriyani²⁾, dan Titik Rahmawati³⁾

^{1, 2,3)}Program Studi Informatika, Fakultas Teknik Universitas Sarjanawiyata Tamansiswa

e-mail: eka.sari@ustjogja.ac.id¹⁾, dina.herijanj@ustjogja.ac.id²⁾, titik.rahmawati@ustjogja.ac.id³⁾

ABSTRAK

Bertambahnya jumlah mahasiswa yang ditampung serta jumlah kelas yang disediakan oleh Fakultas Teknik UST setiap tahunnya akan mengakibatkan proses penyusunan jadwal perkuliahan akan semakin kompleks dan rumit. Saat ini Penyusunan jadwal perkuliahan di Fakultas Teknik UST masih dilakukan secara manual dengan menggunakan excel sehingga proses penyusunan jadwal tersebut membutuhkan waktu yang lama sehingga kurang efektif. Tujuan dari penelitian ini adalah untuk mengatasi permasalahan penjadwalan dengan merancang sebuah model sistem optimasi penjadwalan mata kuliah pada Fakultas Teknik UST. Metode Optimasi yang digunakan untuk penjadwalan adalah algoritma genetika. Pada penelitian ini pemodelan dilakukan pada pemodelan sistem dan pemodelan penyimpanan datanya. Pemodelan proses dari sistem optimasi penjadwalan dengan algoritma genetika dilakukan dengan UML (Unified Modeling Language) yang terdiri dari 3 diagram yaitu Use Case Diagram, Activity Diagram, dan Class Diagram sedangkan pemodelan analisis penyimpanan datanya menggunakan ERD (Entity Relation Diagram). Hasil akhir dari penelitian ini adalah sebuah pemodelan sebagai roadmap dalam pembangunan sistem perangkat lunak optimasi penjadwalan matakuliah dengan algoritma genetika yaitu berupa Use Case Diagram, Activity Diagram, Class Diagram dan ERD (Entity Relation Diagram). Pada penelitian ini menggunakan 4 hard constraint dan 3 soft constraint yang menjadi batasan dalam memproses penjadwalan dengan algoritma genetika.

Kata Kunci: Penjadwalan Matakuliah, Algoritma Genetika, UML (Unified Modeling Language), Hard Constraint, Soft Constraint.

ABSTRACT

The increasing number of students accommodated and the number of classes provided by the UST Faculty of Engineering each year will result in the process of preparing lecture schedules becoming more complex and complicated. At present, the preparation of class schedules at the UST Faculty of Engineering is still done manually using Excel, so the process of preparing the schedule takes a long time, making it less effective. The purpose of this research is to overcome scheduling problems by designing a model of the optimization system for scheduling courses at the UST Faculty of Engineering. The optimization method used for scheduling is a genetic algorithm. In this study, modeling was carried out on system modeling and data storage modeling. Process modeling of the scheduling optimization system with genetic algorithms is carried out using UML (Unified Modeling Language) which consists of 3 diagrams namely Use Case Diagrams, Activity Diagrams, and Class Diagrams while modeling the analysis of data storage using ERD (Entity Relation Diagrams). The result of this research is modeling as a roadmap in the development of a software system for optimizing course scheduling with genetic algorithms, namely Use Case Diagrams, Activity Diagrams, Class Diagrams, and ERD (Entity Relation Diagrams). In this study, 4 hard constraints and 3 soft constraints are used which are the limitations in processing scheduling with genetic algorithms.

Keywords: Course Scheduling, Genetic Algorithm, UML (Unified Modeling Language), Hard Constraint, Soft Constraint.

I. PENDAHULUAN

Penjadwalan mata kuliah di setiap universitas merupakan pekerjaan yang rumit dan kompleks karena banyak kendala dan aturan-aturan yang harus diperhatikan. Kendala tersebut bisa berupa jumlah mata kuliah, jumlah ruangan yang tersedia, jumlah mahasiswa, preferensi dosen

dan slot waktu sehingga permasalahan penjadwalan ini akan sulit dilakukan jika menggunakan metode optimasi konvensional [1]. Kesulitan dalam menerapkan metode optimasi tradisional dalam proses penjadwalan dikarenakan masalah penyusunan jadwal termasuk kedalam golongan *NP-Hard problem* (*nondeterministic polynomial time*) yang

jika semua kombinasi alternatif diuji untuk mengujinya, waktu yang diperlukan untuk mencari solusi akan meningkat tajam [2].

Penjadwalan merupakan hal yang sangat penting dalam proses belajar mengajar karena kegiatan Dosen pengajar dan mahasiswa bergantung pada jadwal perkuliahan. Penyusunan jadwal perkuliahan perlu persiapan yang matang dan mendalam sehingga dikemudian hari tidak mengganggu aktifitas perkuliahan. Penyusunan jadwal perkuliahan memerlukan waktu yang tidak sedikit untuk menyusunnya agar jadwal yang dihasilkan baik dan adil yang memperhatikan batasan-batasan yang ada sehingga proses belajar mengajar berjalan dengan baik, khususnya bagi dosen dan mahasiswa yang mengikuti mata kuliah tersebut.

Apek-aspek yang perlu diperhatikan dalam penjadwalan seperti :1) tidak diperbolehkan terjadi bentrok jadwal, 2) ketersediaan waktu mengajar dosen, 3) keterbatasan ruangan dan 4.) distribusi mata kuliah harus merata setiap minggunya [3]. Permasalahan umum yang perlu diperhatikan pada saat penjadwalan yaitu kemungkinan dosen mengajar lebih dari satu mata kuliah dan kemungkinan jumlah mata kuliah dan dosen tidak seimbang.

Fakultas Teknik Universitas Sarjanawiyata Tamansiswa (UST) merupakan salah satu fakultas di universitas swasta di Yogyakarta. FT UST memiliki 3 program studi yang meliputi Teknik Sipil, Teknik Industri, dan Informatika. Setiap awal semester, Tim dari Fakultas Teknik UST akan memetakan matakuliah, dosen, mahasiswa, waktu dan ruangan yang digunakan untuk membuat jadwal perkuliahan. Penyusunan jadwal perkuliahan di Fakultas Teknik UST masih dilakukan secara manual dengan menggunakan excel sehingga proses penyusunan jadwal tersebut membutuhkan waktu yang lama sehingga kurang efektif. Sistem optimasi penjadwalan matakuliah menjadi solusi dengan memanfaatkan optimasi penjadwalan menggunakan algoritma. Berbagai algoritma optimasi telah digunakan oleh berbagai peneliti, salah satunya Algoritma Genetika [4][5]. Algoritma genetika merupakan algoritma optimasi yang bekerja meniru proses seleksi alam yang dapat menyelesaikan permasalahan optimasi kompleks dan memiliki model pencarian yang luas.

Berdasarkan latar belakang masalah tersebut, peneliti tertarik untuk merancang sebuah pemodelan sistem optimasi penjadwalan matakuliah di Fakultas Teknik UST dengan algo-

ritma Genetika untuk mempermudah pengembangan sistem penjadwalan sehingga mengoptimalkan waktu penyusunan jadwal matakuliah. Data yang diperoleh merupakan data sekunder berupa data penjadwalan pada Fakultas Teknik UST semester genap tahun ajar 2022/2023.

II. STUDI PUSTAKA

Ferdinandus (2021) mendefinisikan penjadwalan mata kuliah sebagai aktivitas pengisian kegiatan perkuliahan yang dilaksanakan selama seminggu dalam satu hari. Perkuliahan yang terdiri dari dosen, mata kuliah, mahasiswa, ruang kuliah dan rencana pelaksanaannya diatur sedemikian rupa sehingga tidak terjadi bentrok selama perkuliahan maka jadwal perkuliahan ini menjadi sangat penting sehingga rutin dilakukan setiap semester. Membuat rencana perkuliahan dalam sebuah susunan jadwal yang baik pasti akan meningkatkan kualitas dan pelayanan pendidikan [6].

Telah banyak penelitian tentang optimasi penjadwalan dengan menggunakan algoritma yang berbeda-beda, seperti yang dilakukan oleh Agus Rudianto dkk (2022) dalam penelitiannya yang berjudul “Rancang Bangun System Penyusunan Jadwal Pelajaran Menggunakan Algoritma Genetika Berbasis Web” [7]. Pada penelitian ini menerapkan algoritma genetika untuk mengoptimalkan pembuatan jadwal pelajaran dalam kegiatan administrasi sekolah dengan memanfaatkan sistem berbasis web. Penelitian ini menemukan kesimpulan bahwa penggunaan algoritma genetika memberikan penjadwalan pelajaran yang cukup baik dan dapat mengoptimalkan penugasan belajar mengajar kepada guru bersertifikat atau tidak bersertifikat.

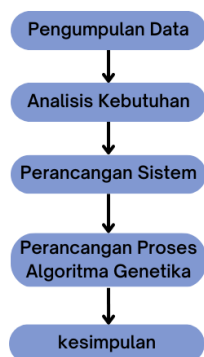
Anggi Andriyadi dkk (2022) juga melakukan penelitian dengan judul “Penerapan Algoritma Genetika untuk Penjadwalan Sidang dan Seminar IIB Darmajaya” [8]. Penelitian ini bertujuan untuk membuktikan penerapan algoritma genetika pada sistem penjadwalan seminar proposal dan sidang skripsi secara otomatis apakah lebih optimal. Hasil dari penelitian ini yaitu dapat disimpulkan bahwa sistem penjadwalan yang dibangun menggunakan algoritma genetika dapat mempercepat proses kegiatan penjadwalan dan mempermudah mahasiswa dan dosen untuk mengetahui jadwal seminar proposal dan sidang skripsi. Penerapan algoritma genetika pada sistem yang dibangun pada penelitian ini yaitu meminimalisir kesalahan dan keterlambatan proses penjadwalan.

Pada penelitian yang berjudul “Optimalisasi Sistem Penjadwalan Mata Kuliah Jurusan Bahasa Inggris Menggunakan Algoritma Genetika (Studi kasus Politeknik Negeri Surabaya)” yang dilakukan oleh rasmalia dkk(2022)[9]. Hasil akhir dari penelitian ini yaitu dengan menggunakan sistem berbasis web, proses penjadwalan matakuliah semakin optimal dan tidak ada bentrok antar mata kuliah , antar dosen dengan yang lain, serta antar jam dan waktu perkuliahan.

Penelitian lain yang berjudul “Membangun Sistem Informasi Penjadwalan dengan Metode Algoritma Genetika pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara” yang dilakukan oleh Agil Assagaf dkk (2018)[10]. Pada penelitian ini menggunakan metode algoritma genetika untuk optimasi dalam pembuatan jadwal praktikum. Menurut penelitian ini, Parameter-parameter yang mempengaruhi proses penjadwalan perkuliahan adalah jumlah individu, probabilitas crossover, probabilitas mutase serta metode seleksi.

III. METODE PENELITIAN

Pada penelitian ini, tahapan yang digunakan terlihat pada gambar 1 berikut ini.



Gambar 1 Tahap Penelitian

A. Pengumpulan data

Tahap ini dilakukan untuk menggali data apa saja yang digunakan untuk pembuatan sistem dan analisis algoritma GA dalam pembuatan sistem penjadwalan perkuliahan. Pengumpulan data dilakukan dengan metode observasi dan wawancara dengan tim penyusun jadwal perkuliahan Fakultas Teknik UST. Tim penyusun jadwal tersebut membutuhkan waktu yang lama dalam menyusun jadwal perkuliahan sehingga tidak efektif dan efisien. Terkadang ada tabrakan jadwal antar matakuliah lain ataupun tabrakan jadwal dosen dalam mengajar. Hasil obeservasi dan wawancara didapatkan data sekunder yang bentuknya berupa data penjadwalan perkuliahan Fakultas Teknik

UST semester Genap 2022/2023 yang terdiri dari data mahasiswa, matakuliah, dosen, ruangan, dan sesi perkuliahan serta aturan-aturan dalam penjadwalan. Data dosen yang diperoleh adalah 45 data, data ruang sebanyak 9 ruang, data mata kuliah sebanyak 66 mata kuliah, hari aktif perkuliahan sebanyak 5 hari, dan ada 5 sesi setiap harinya.

B. Analisis Kebutuhan

Kendala yang dipertimbangkan pada penilitian ini merujuk pada [11] yaitu mengatur jadwal dengan memastikan bahwa mata kuliah, dosen, mahasiswa, kelas, ruang, hari, dan sesi tidak terjadi bentrok satu sama lainnya disetiap periode kuliah dan memastikan bahwa telah memenuhi batasan yang ditetapkan oleh administrator sebelumnya. Pada paper ini, mempertimbangkan aturan sebagai berikut :

- Tidak boleh ada tabrakan jadwal antar matakuliah
- Tidak boleh ada tabrakan jadwal antar dosen.
- Tidak boleh ada tabrakan jadwal antar kelas yang diambil mahasiswa
- Hanya 1 dosen yang mengampu 1 matakuliah dalam 1 kelas yang dapat mengisi ruangan dalam 1 sesi.
- Tidak boleh tabrakan jadwal antar kelas teori dan praktik.
- Jadwal kelas praktikum sudah di tetapkan sebelumnya.

Peneliti juga memetakan analisis kebutuhan fungsional sistem yang mengacu pada kebutuhan sistem yang diusulkan. Pada sistem penjadwalan dengan algoritma genetika ini memiliki 1 user utama yaitu administrator. Kebutuhan fungsional telah dipetakan dalam tabel 1 berikut ini :

Tabel 1. Kebutuhan Fungsional User

No	Kebutuhan Utama	Tujuan
1	Mengelola data matakuliah, ruangan, sesi, mahasiswa dan dosen	Melakukan proses CRUD data sehingga data yang diinputkan ke sistem dipastikan sudah benar.
2	Mengelola data pengampu matakuliah	Melakukan proses setting pengampu matakuliah
3.	Mengelola hasil penjadwalan disetiap semester	Melakukan download jadwal yang sudah berhasil di susun oleh sistem
4.	Mengelola data matakuliah praktikum	Melakukan setting data matakuliah praktikum yang sudah didapatkan

dari jadwal penggunaan
Laboratorium UST

Setelah dipetakan kebutuhan fungsional user, selanjutnya ke tahap perancangan sistem.

C. Perancangan Sistem.

Pada tahap perancangan ini dilakukan dengan membuat rancangan sistem dengan sebuah UML (*Unified Modeling Language*). UML sebagai alat untuk memodelkan sistem yang berorientasi objek yang menjadikan proses analisis dan design dalam tahap iteratif. Tahap *iterative* tersebut meliputi : identifikasi kelas dan objek, identifikasi hubungan antar objek dan kelas serta perincian *interface* dan implementasinya[12]. Pada penelitian ini akan menggunakan 3 diagram yaitu *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram* sedangkan pemodelan basis datanya menggunakan ERD (*Entity Relation Diagram*).

D. Perancangan Proses Algoritma Genetika.

Penelitian ini menggunakan algoritma genetika untuk proses optimasi penjadwalan. Algoritma genetika merupakan algoritma pencarian numerik yang didasarkan pada cara kerja dari seleksi alam dan genetika makhluk hidup. Algoritma ini biasa digunakan untuk menyelesaikan permasalahan yang berhubungan dengan masalah pencarian solusi optimal[13]. Mencari solusi optimal melalui proses iteratif untuk membentuk individu dari populasi yang merupakan individu bernilai tinggi. Solusi optimal dihasilkan dengan mencari di antara titik-titik optimal alternatif berdasarkan fungsi probabilitas. Salah satu kelebihan dari algoritma genetika yaitu sifatnya yang mencari solusi global sehingga tidak mudah terjebak dalam optimasi lokal[14]. Prosedur algoritma genetika mengikuti langkah-langkah yang mirip dengan proses evolusi yang terdiri dari proses seleksi, crossover dan mutasi.

Awalnya gen dikodekan dalam bentuk kromosom sehingga dapat diproses oleh algoritma genetika. Perlu digaris bawahi bahwa setiap kromosom mengandung beberapa gen yang mengkodekan informasi yang disimpan dalam individu atau kromosom. Kemudian menginisialisasi populasi awal dengan membangkitkannya secara acak untuk mendapatkan solusi awal. Suatu populasi terdiri dari individu-individu dengan kromosom yang terdiri dari sekumpulan gen. Selanjutnya dilakukan evaluasi populasi untuk menentukan nilai fitness

dari setiap kromosom sampai terpenuhi kriteria berhenti menggunakan persamaan (1) berikut :

$$fitness = \sum_{i=1}^n G_i P_i \quad (1)$$

Keterangan :

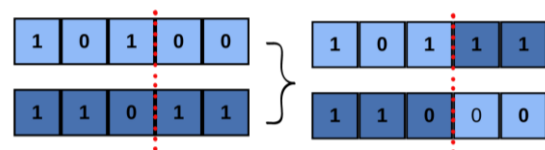
Fitness : Nilai fitness dari setiap kromosom

G_i : Nilai gen ke-i pada kromosom yang dievaluasi.

P_i : Nilai prioritas perangkat ke-i

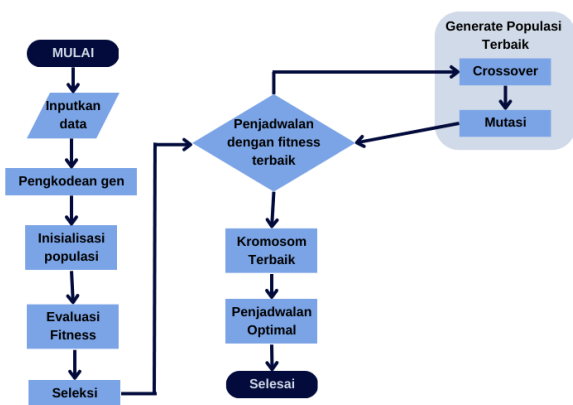
Selanjutnya, kumpulan kromosom dalam populasi dilakukan seleksi dengan memilih kandidat kromosom berdasarkan nilai fitness terbaik. Kandidat kromosom yang terpilih akan digunakan untuk proses perkawinan atau persilangan. Proses selanjutnya adalah *crossover* atau persilangan, dimana informasi dipertukarkan antara dua kromosom yang berbeda untuk menciptakan kromosom baru yang memiliki karakteristik dari kedua kromosom tersebut. Penelitian ini menggunakan *single-point crossover*, dimana setiap gen pada satu kromosom dipertukarkan dengan gen pada kromosom lain untuk membuat kromosom baru melalui satu titik potong.

[15]. Titik potong tersebut dibangkitkan menggunakan sebuah *random number* dengan batasan 1 sampai n (panjang kromosom). Contohnya pada kasus 2 kromosom yang memiliki panjang 5 serta random number yang didapatkan adalah 3, maka gen 1 sampai 3 akan dipotong dengan gen 4 sampai 5. Kemudian gen 1 sampai 3 pada kromosom pertama akan dilakukan crossover dengan gen 4 dan 5 pada kromosom dua, dan sebaliknya. Ilustrasi proses crossover menggunakan crossover one point crossover terlihat pada gambar 3 berikut ini.



Gambar 2 Ilustrasi *One Point Crossover*

Setelah dilakukan persilangan, kromosom akan melalui proses mutasi untuk mengubah nilai dari suatu gen pada kromosom tersebut. Perubahan nilai dari satu atau lebih gen pada kromosom tersebut dilakukan secara random dengan interval area pencarian. Tujuan proses mutasi adalah membentuk individu-individu dengan kualitas terbaik diatas rata-rata serta untuk memperbaiki kerusakan gen akibat proses crossover[16]. Pada gambar 3 dibawah merupakan tahap-tahap optimasi penjadwalan dengan algoritma genetika :

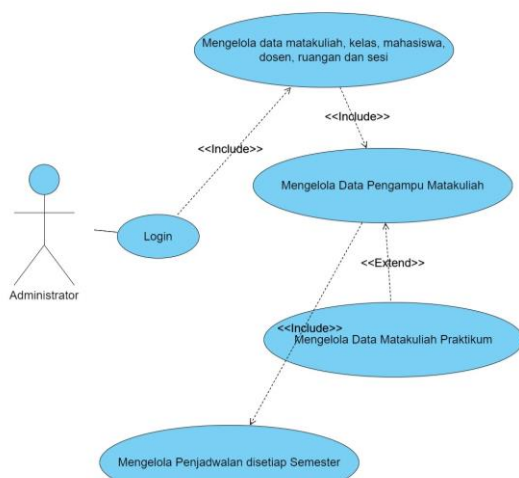


Gambar 3 Tahap-tahap Proses Penjadwalan dengan Algoritma Genetika

IV. HASIL DAN PEMBAHASAN

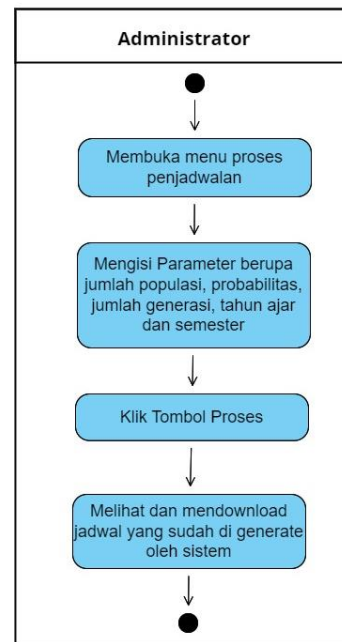
A. Perancangan Sistem

Penulis merancang use case diagram yang digunakan untuk memodelkan perilaku sistem dengan mendefinisikan fitur apa saja yang dimiliki sistem dan siapa yang berhak menggunakan fitur tersebut[17]. Berikut ini rancangan use case diagram dari sistem penjadwalan matakuliah dengan algoritma genetika pada Fakultas Teknik UST.



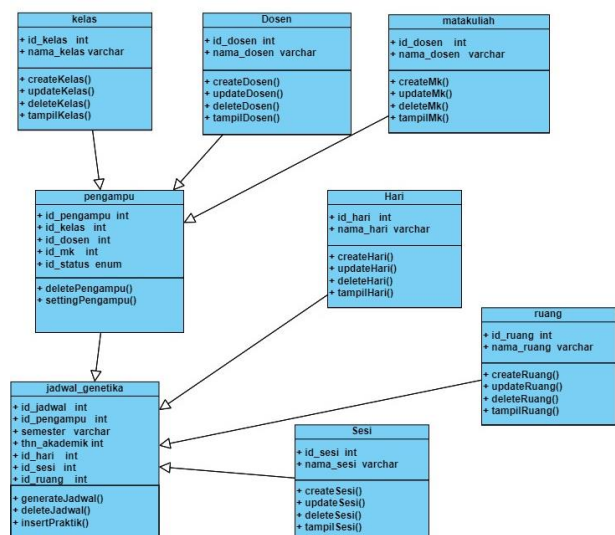
Gambar 4 Use Case Diagram Sistem Penjadwalan Matakuliah dengan GA

Setelah *Use Case Diagram* dibuat, Kemudian buat skenario dengan *Activity diagram* untuk setiap use case diagram. *Activity Diagram* digunakan untuk menggambarkan sistem kerja atau sistem suatu objek secara terstruktur. Pada artikel ini, penulis menambahkan *activity diagram* pada saat pembentukan jadwal perkuliahan setiap semester seperti yang ditunjukkan pada Gambar 5 di bawah ini



Gambar 5 Activity Diagram pada Proses Mengelola Penjadwalan

Selanjutnya, penulis membuat sebuah logical model dari sistem dengan *Class Diagram*. *Class diagram* berfungsi untuk memudahkan dalam menunjukkan bagaimana skema dari arsitektur sistem yang dirancang[18]. Class diagram sendiri merupakan alat untuk menggambarkan sistem dalam bentuk kelas yang mencakup status dan perilaku dalam hubungan antar kelas. Berikut ini rancangan class diagram dari sistem optimasi penjadwalan matakuliah dengan algoritma genetika yang diusulkan oleh penulis :

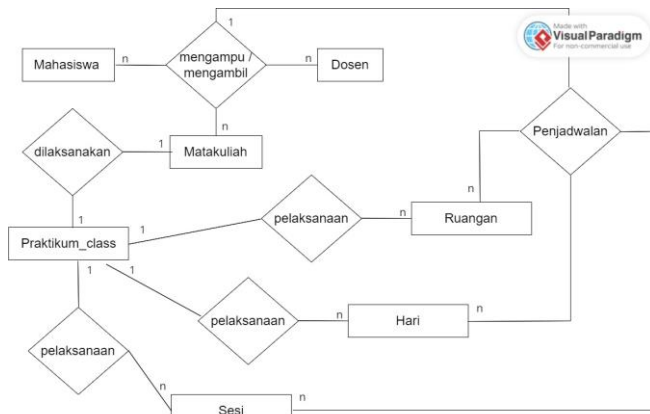


Gambar 6 Class Diagram Sistem Penjadwalan Matakuliah dengan GA

B. Perancangan ERD (Entity Relation Diagram)

Entity Relation Diagram atau ERD digunakan penulis untuk merancang database yang digunakan dalam sistem optimasi penjadwalan matakuliah dengan algoritma genetika. Pada ERD akan di

deskripsikan data yang akan disimpan dalam sistem maupun batasannya. Komponen utama pada diagram ERD pada umumnya adalah entity set, relationship set, dan *constraints*[19]. Berikut rancangan ERD yang digunakan pada penelitian ini, terlihat pada gambar 7 berikut :



Gambar 7 Entity Relation Diagram Sistem Optimasi Penjadwalan dengan GA

C. Perancangan Penjadwalan dengan Algoritma Genetika

Penjadwalan dilakukan untuk mengalokasikan rangkaian kejadian terbatas sumber daya yang diatur untuk memenuhi kendala tertentu. Batasan tersebut dapat diklasifikasikan menjadi *Hard Constraint* (harus dipenuhi) dan *Soft Constraint* (harus dipenuhi). *Hard Constraint* berarti bahwa kendala ini harus diterapkan dan dipenuhi saat mencari solusi penjadwalan yang optimal. Pencarian solusi dianggap valid jika tidak ada hard constraint pencarian solusi pada jadwal yang dilanggar.

Pada penelitian ini, terdapat beberapa batasan dalam proses pencarian jadwal optimal dengan algoritma genetika yaitu :

1. Hard Constraint

- Tidak boleh ada bentrok dosen terhadap sesi waktu mengajar.
- Tidak boleh ada bentrok antar dosen pada ruang kuliah tertentu.
- Tidak boleh ada bentrok antar mahasiswa terhadap sesi waktu kuliah dan ruangan kuliah.
- Tidak boleh ada bentrok matakuliah teori dan praktikum

2. Soft Constraint

- Diupayakan untuk memberi jeda 1 sesi jika dalam sehari ada kelas teori dan praktikum.
- Diupayakan untuk mengkosongkan sesi pukul 11.00-13.00 hari jumat kosong.
- Diupayakan untuk setiap kelasnya maksimal 4 sesi perkuliahan

Langkah-langkah pada algoritma genetika yang digunakan pada perancangan sistem optimasi penjadwalan matakuliah ini adalah sebagai berikut :

A. Inisialisasi Kromosom

Pada langkah ini, data yang ada dibentuk menjadi variabel yang diperlukan untuk pembentukan kromosom. Variabel ini berupa: kode pengampu mk, kode ruang, kode hari, dan kode sesi. Variabel kode pengampu terdiri dari kode dosen, kode MK dan kelas. Variabel dosen, kode_mk dan kelas disebut gen dan gabungan variabel kode_mk, kelas, hari dan kode sesi disebut kromosom. Setiap kromosom memiliki solusi, sehingga semakin banyak kromosom yang dibuat, semakin banyak solusi yang mungkin.

Pada tabel berikut merupakan ilustrasi gen dan kromosom.

Tabel 2. Ilustrasi Gen pada Optimasi Penjadwalan Algoritma Genetika

kode_pengampu			kd_hari	kd_sesi	kd_ruang
kd_dosen	kd_mk	cls			
D1	M1	K1	1	S1	R1
D2	M2	K2	2	S2	R2
D3	M3	K3	3	S3	R3
D4	M4	K4	4	S4	R4
D5	M5	K5	5	S5	R5
D6	M6	K6		S6	R6
$D_n \dots$	$M_n \dots$	$K_n \dots$			R7

$$kromosom = [P_{kp}, H_{kh}, S_{ks}, R_{kr}] \quad (2)$$

Pada persamaan 2 diatas menggambarkan bahwa pada setiap kromosom terdapat pengampu mengajar pada hari ke- kh , sesi ke- ks , dan ruang ke- kr .

a. Inisialisasi populasi awal

Pada langkah ini, populasi dibangkitkan secara acak hingga membentuk populasi awal. Berikut *pseudocode* pembangkitan populasi awal yang terlihat pada Gambar 8 berikut ini :

```

PopulasiAwal()
Inisialisasi :
    jml_pengampu, jml_sesi, jml_hari, jml_ruang = count(ruangteori)-count(ruangpraktikum); populasi = 0; j=0;
    for i=1 to populasi, j=1;
        for j=1 to jml_pengampu, j=1;
            if status[j] = 0 then
                j=1;
            } else if status[j] = 1 then
                populasi = j;
            }
        Inisialisasi :
            individu[i][j][0]=0;
            if (cekPraktikumClassroom)
                then
                    individu[i][j][1]=rand(0, jml_sesi-1);
                    individu[i][j][2]=rand(0, jml_hari-1);
                    individu[i][j][3]=rand(0, jml_ruang-1);
            } else if (cekPraktikumClassroom)
                then
                    Inisialisasi :
                        id_pengampu, id_mk, id_ruangPraktikum, id_sesiPraktikum, id_hariPraktikum
                        insert to tabel
                        jadwal_genetis (id_pengampu, id_mk, id_ruangPraktikum,
                            id_sesiPraktikum, id_hariPraktikum);
                    populasi = j;
            }
    }

```

Gambar 8 Pseudocode Pembangkitan Populasi Awal

Gambar 8 diatas merepresentasikan tahapan bagaimana proses pembangkitan popuasi awal dilakukan. Inisialisasikan dahulu variabel jumlah seluruh pengampu, jam, hari dan ruang yang akan digunakan pada proses pejadwalan. Kemudian acak individu secara random berdasarkan banyaknya sesi, hari dan ruang sehingga terbentuk sebuah array.

b. Menghitung Fitness

Tahap ini merupakan tahap menghitung nilai fitness dengan mempertimbangkan sebuah batasan-batasan yang sudah ditentukan (hard constraint). Nilai fitness yang diharapkan adalah yang memiliki nilai yang paling besar yaitu yang memiliki jumlah pelanggaran paling minimum diantara yang lain. Pada gambar 9 berikut ini merupakan *pseudocode* proses cek nilai *fitness*.

```

cekFitness{
  inisialisasi :
  pinalti0, count_pemampu, i=0;
  for i<-count_pemampu; i++{
    then
      inisialisasi :
      sesi_a, hari_a, ruang_a, dosen_a;
      hari_praktikum[], sesi_praktikum[], jml_mPraktikum=countPraktikum();
      for i<-jml_pemampu; i++{
        inisialisasi :
        sesi_b, hari_b, ruang_b, dosen_b;
        if i=j then continue;
        if sesi_a==sesi_b & hari_a==hari_b & ruang_a = ruang_b
          then increment pinalti[i];
        if sesi_a==sesi_b & hari_a==hari_b & dosen_a==dosen_b
          then increment pinalti[i];
        for p<-jmlPraktikum; p++{
          if sesi_praktikum[p]==sesi_a & hari_praktikum[p]==hari_a{
            then increment pinalti[i];
          }
        }
      }
    }
  }
  fitness=1/(pinalti);
  return fitness;
}

```

Gambar 9 *Pseudocode* Cek Fitness

Batasan-batasan yang digunakan untuk menghitung nilai fitness yaitu :

- Tidak boleh ada bentrok dosen terhadap sesi waktu mengajar
- Tidak boleh ada bentrok antar dosen pada ruang kuliah tertentu.
- Tidak boleh ada bentrok antar mahasiswa terhadap sesi waktu kuliah dan ruangan kuliah.
- Tidak boleh ada bentrok matakuliah teori dan praktikum

Jika batasan-batasan tersebut dilanggar, maka akan menambah variabel pinalti. Dimana variabel pinalti ini digunakan untuk menghitung nilai fitness masing-masing individu.

c. Seleksi

Pada tahap ini, dilakukan proses seleksi dengan metode seleksi rangking. Pada metode seleksi rangking ini, nilai fitness terkecil diberi 1, yang terkecil kedua bernilai 2 dan begitu seterusnya sampai yang terbesar, dengan demikian semakin besar nilai fitness maka rangkingnya semakin besar. Nilai tersebut akan diambil sebagai prosentase tepat yang tersedia. Pada gambar 10 berikut ini merupakan *pseudocode* proses seleksi.

```

prosesSeleksi(fitness){
  inisialisasi :

  jumlah=0, rangking[], populasi,i=0;
  for i<-populasi; i++{
    inisialisasi :
    rangking[i]=1,j=0;
    for j<-populasi{
      inisialisasi :
      fitness_a=fitness[i],fitness_b=fitness[j];
      if fitness_a>fitness_b
        then increment rangking[i]++;
      inisialisasi :

    }
    jumlah +=rangking[i];
  }
  inisialisasi :
  jml_rang=count(rangking),p=0;
  for p<-populasi;p++{
    inisialisasi :
    target=random (0,jumlah-1),cek=0,jr=0;
    for jr<-jml_rang ; jr++{
      increment cek += rangking[jr];
      if cek>=target
        then induk[i]=jr;
        break;
    }
  }
}

```

Gambar 10 *Pseudocode* Proses Seleksi

Nilai fitness dari suatu kromosom menunjukan kualitas dari kromosom dalam populasi tersebut. Kromosom yang memiliki nilai fitness terkecil akan menjadi kromosom terbaik dan mempunyai peluang yang besar untuk dipilih menjadi induk (parent).

d. Crossover

Pada proses seleksi sebelumnya, akan didapatkan parents yang akan digunakan untuk proses crossover. Pada penelitian ini menggunakan *crossover one point crossover*. Proses crossover sendiri tidak akan terlepas dari parameter algoritma genetika yaitu probabilitas crossover. Parameter probabilitas ini digunakan untuk mengontrol setiap operator crossover. Gen-gen dalam populasi mengalami persilangan yang mana semakin tinggi probabilitas persilangan, semakin cepat terbentuk struktur tunggal baru dalam populasi. Jika probabilitas silang terlalu tinggi, satu kandidat solusi yang paling mungkin akan menghilang lebih cepat dalam proses pembangkitan ulang. Pada gambar 11 terlihat pseudocode proses crossover.

```

crossover{
  inisialisasi:
  individu_baru[j], jml_pengampu;
  for i=0, i<populasi; i++{
    inisialisasi:
    cr= random(0, maxrandom-1)/maxrandom;
    if cr<probabilitas crossover
      then
        inisialisasi:
        a= random(0, jml_pengampu-1);
        for j=0, j<a, j++{
          for k=0, k<4, k++{
            individu_baru[i][j][k]=individu[i][j][k];
            individu_baru[i+1][j][k]=individu[i+1][j][k];
          }
        }
        for j=a, j<jml_pengampu, j++{
          for k=0, k<4, k++{
            individu_baru[i][j][k]=individu[i][j][k];
            individu_baru[i+1][j][k]=individu[i+1][j][k];
          }
        }
      }
    else then
      for j=0, j<jml_pengampu, j++{
        for k=0, k<4, k++{
          individu_baru[i][j][k]=individu[i][j][k];
          individu_baru[i+1][j][k]=individu[i+1][j][k];
        }
      }
    }
  }
  inisialisasi:
  jumlah_pengampu=count(pengampu);
  for i=0, i<jumlah_pengampu, i++{
    for j=0, j<4, j++{
      for k=0, k<4, k++{
        individu[i][j][k]=individu_baru[i][j][k];
        individu[i+1][j][k]=individu_baru[i+1][j][k];
      }
    }
  }
}

```

Gambar 11 Pseudocode Proses Crossover

Pada proses *crossover* ini dilakukan perulangan dan apabila nilai acak *crossover* pada populasi kurang dari nilai probabilitas *crossover* maka akan terjadi pertukaran jadwal. Perulangan terus dilakukan sampai nilai acak bisa melebihi probabilitas *crossover*.

e. Mutasi

Pada tahap mutasi, hal ini dilakukan dengan mengacak ulang komponen penjadwalan. Ketika proses mutasi selesai, skor fitness masing-masing individu dihitung. Jika tidak ada yang cocok dengan batasan-batasan yang ditentukan, proses mutasi diulang. Proses mutasi tergambar pada pseudocode yang terlihat pada gambar 12 berikut :

```

prosesMutasi{
  inisialisasi:
  fitness[];
  r= random(0, randommax-1)/randommax;
  jml_pengampu, jml_sesi, jml_hari, jml_ruang;
  for i=0, i<populasi, i++{
    if r> mutasi
      then
        krom= random(0, jml_pengampu-1);
        individu[i][krom][1]=random(0, jml_sesi-1);
        individu[i][krom][2]=rand(0, jml_hari-1);
        individu[i][krom][3]=rand(0, jml_ruang-1);
        fitness[i]=function cekFitness(i);
      }
  }
  return fitness;
}

```

Gambar 12 Pseudocode Proses Mutasi

Gambar 12 menjelaskan *pseudocode* proses mutasi pada algoritma genetika. Proses mutasi dilakukan dengan mengacak kembali penjadwalan untuk jadwal baru yang terbentuk. Kemudian dilakukan perulangan untuk populasi. Jika nilai acak kurang dari nilai probabilitas mutasi maka komponen penjadwalan akan diganti. Langkah terakhir

yaitu mengecek kembali nilai *fitness* setiap individu dengan probabilitas mutasi.

V. KESIMPULAN

Berdasarkan hasil pembahasandiatas maka pada penelitian ini dapat diambil kesimpulan sebagai berikut :

1. Masalah penjadwalan matakuliah pada Fakultas Teknik Universitas Sarjanawiyata Tamansiswa dapat dimodelkan sebagai pemodelan sistem optimasi penjadwalan matakuliah dengan algoritma genetika.
2. Menghasilkan pemodelan sistem optimasi penjadwalan matakuliah yang dapat ditindak lanjuti dengan mengimplementasikan model pada penelitian ini sebagai roadmap dalam pembangunan sistem perangkat lunak optimasi penjadwalan matakuliah dengan algoritma genetika.
3. Pada pemodelan sistem optimasi penjadwalan ini menghasilkan 4 *hard constraint* dan 3 *soft constraint*.
4. Penelitian ini menghasilkan pemodelan aliran dan proses sistem dengan UML (*Unified Modeling Language*) yang terdiri dari diagram yaitu *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Sedangkan pemodelan basis data untuk sistem optimasi penjadwalan dengan Genetika menggunakan ERD (*Entity Relation Diagram*).

DAFTAR PUSTAKA

- [1] H. Kanoh and Y. Sakamoto, "Knowledge-based genetic algorithm for university course timetabling problems," *Int. J. Knowledge-Based Intell. Eng. Syst.*, vol. 12, no. 4, pp. 283–294, 2008, doi: 10.3233/KES-2008-12403.
- [2] T. Birbas, S. Daskalaki, and E. Housos, "School timetabling for quality student and teacher schedules," *J. Sched.*, vol. 12, no. 2, pp. 177–197, 2009, doi: 10.1007/s10951-008-0088-2.
- [3] H. U. A. Hartadi Rudy and V. G., "Perancangan Aplikasi Penjadwalan Mata Kuliah," *J. Bianglala Inform.*, vol. 4, no. 1, pp. 31–40, 2016, doi: 10.4324/9781003221296-10.
- [4] H. P. Hariyadi, T. Widiyaningtyas, M. Z. Arifin, and S. Sendari, "Implementation of Genetic Algorithm to academic scheduling system," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 2013–2016, 2017, doi: 10.1109/TENCON.2016.7848378.
- [5] X. Chen, R. Yi, and M. Li, "Design and Application of an Improved Genetic Algorithm to a Class Scheduling System," pp. 44–59.
- [6] F. Mone and J. E. Simarmata, "Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 15, no. 4, pp. 615–628, 2021, doi: 10.30598/barekengvol15iss4pp615-628.
- [7] A. Rudianto and I. Muhandhis, "Rancang Bangun Sistem Penyusunan Jadwal Pelajaran Menggunakan Algoritma Genetika Berbasis Web (study Kasus MI Mahalul Ulum)," *J. Syst. Eng. Technol. Innov.*, vol. 1, no. 01, pp.

- 33–37, 2022, doi: 10.38156/jisti.v1i01.14.
- [8] H. Ardiansyah and M. B. S. Junianto, “Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran,” *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 329, 2022, doi: 10.30865/mib.v6i1.3418.
 - [9] S. D. Kristanti, P. Studi, T. Informatika, U. B. Darma, and K. Palembang, “Optimalisasi Sistem Penjadwalan Mata Kuliah Jurusan Bahasa Inggris Menggunakan Algoritma Genetika (Studi Kasus Politeknik Negeri Sriwijaya),” vol. 14, no. 2, pp. 22–31, 2022.
 - [10] C. Suranto, “Membangun Sistem Informasi Penjadwalan dengan Metode Algoritma Genetika pada Laboratorium Teknik Informatika UMMU Ternate,” *J. Ilm. Ilmu Komput. dan Inform.*, vol. 1, no. 2, p. 65, 2018.
 - [11] T. Li, Q. Xie, and H. Zhang, “Design of College Scheduling Algorithm Based on Improved Genetic Ant Colony Hybrid Optimization,” *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/2565639.
 - [12] A. Voutama, “Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML,” *Komputika J. Sist. Komput.*, vol. 11, no. 1, pp. 102–111, 2022, doi: 10.34010/komputika.v11i1.4677.
 - [13] R. N. Kurniawan *et al.*, “Penjadwalan Penggunaan Perangkat Elektronik Menggunakan Algoritma Genetika,” vol. 9, no. 5, pp. 2167–2174, 2022.
 - [14] J. Y. Setiawan, D. E. Herwindiati, and T. Sutrisno, “Algoritma Genetika Dengan Roulette Wheel Selection dan Arithmetic Crossover Untuk Pengelompokan,” *J. Ilmu Komput. dan Sist. Inf.*, vol. 7, no. 1, pp. 58–64, 2019, [Online]. Available: <https://journal.untar.ac.id/index.php/jiksi/article/view/5882>
 - [15] J. Suryaputra, C. Lubis, and T. Sutrisno, “Pemilihan Crossover pada Algoritma Genetika Untuk Program Aplikasi Pengenalan Karakter Tulisan Tangan,” *J. Ilmu Komput. dan Sist. Inf.*, vol. 6, no. 1, pp. 69–72, 2018, [Online]. Available: <https://journal.untar.ac.id/index.php/jiksi>
 - [16] I. Hidayat, S. Revo, L. Inkiriwang, and P. A. K. Pratas, “Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika Pada Proyek Rehabilitasi Puskesmas Minanga,” *J. Sipil Statik*, vol. 7, no. 12, pp. 1669–1680, 2019, [Online]. Available: <https://ejournal.unsrat.ac.id/v3/index.php/jss/article/view/26145>
 - [17] Ade Hendini, “PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK),” *J. KHATULISTIWA Inform.*, vol. IV, no. 2, pp. 107–116, 2016, doi: 10.1145/358315.358387.
 - [18] W. Aliman, “PERANCANGAN PERANGKAT LUNAK UNTUK MENGGAMBAR DIAGRAM BERBASIS ANDROID,” *Syntax Lit. J. Ilm. Indones.*, vol. 6, no. 6, pp. 3091–3098, 2021, [Online]. Available: <https://jurnal.syntaxliterate.co.id/index.php/syntax-literate/article/view/1404>
 - [19] M. Larassati, A. Latukolan, A. Arwan, and M. T. Ananta, “Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram Ke Dalam Database,” *urnal Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, p. 4059, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>