

APLIKASI ALGORITMA GENETIKA DALAM PENJADWALAN MATA KULIAH

Application of Genetic Algorithm in Scheduling Subjects

Ferdinandus Mone^{1*}, Justin Eduardo Simarmata²

^{1,2} Program Studi Pendidikan Matematika, FIP, Universitas Timor
Jl. Eltari, Km. 09, Kel. Sasi, Kota Kefamenanu NTT, 85613, Indonesia

Corresponding author e-mail: ^{1*} ferdimone@gmail.com

Abstrak

Pembuatan jadwal kuliah dapat menjadi masalah dan menghabiskan waktu yang cukup lama apabila terdapat beberapa kendala seperti minimnya ruang kuliah, minimnya tenaga pengajar, dan jumlah mata kuliah yang tersedia dalam satu semester cukup banyak. Penelitian ini bertujuan mengaplikasikan algoritma genetika dalam pembuatan jadwal kuliah sehingga mempermudah proses pembuatan jadwal kuliah. Metode yang digunakan adalah metode waterfall dengan tahapan *Software Development Life Cycle*. Hasil penerapan aplikasi genetika menunjukkan bahwa proses pembuatan jadwal mata kuliah dapat mengatasi *constraint* 1) bentrok ruang dan waktu, 2) bentrok dosen, 3) bentrok jam sholat jumat, 4) terdapat waktu keinginan dosen karena alasan tertentu, dan 5) praktikum pada ruang laboratorium. Dengan melewati *constraint* tersebut, maka aplikasi algoritma genetika pada penjadwalan mata kuliah dikategorikan efektif. Berdasarkan hasil *running* pada 51 dosen pengampu mata kuliah (51 kromosom) diperoleh rata-rata waktu *running* sebanyak 30 kali berturut-turut adalah 25,86 menit sehingga penggunaan aplikasi algoritma genetika dalam penjadwalan mata kuliah efisien.

Kata Kunci : algoritma genetika, penjadwalan, mata kuliah, optimisasi

Abstract

Making a class schedule becomes problem and takes a long time because of several obstacles such as the lack of lecture rooms, the lack of teaching staff, and the high of courses available in one semester. This study aims to apply genetic algorithms in making class schedules to facilitate the process of making class schedules. The method used is the waterfall method with the stages of the *Software Development Life Cycle*. The results of the application of genetics application show that the process of making course schedules can overcome the constraints of 1) space and time clashes, 2) lecturer conflicts, 3) Friday prayer times clashing, 4) there is a time when the lecturer wants for certain reasons, and 5) practicum in the laboratory room. By passing these constraints, the application of genetic algorithms in course scheduling is categorized as effective. Based on the results of running on 51 lecturers (51 chromosomes), the average running time 30 times in a row is 25.86 minutes so that the use of genetic algorithm applications in scheduling courses is efficient.

Keywords: genetic algorithm, scheduling, lecture, optimization

Article info:

Submitted: 12th May 2021

Accepted: 17th October 2021

How to cite this article:

F. Mone, and J. E. Simarmata, "APLIKASI ALGORITMA GENETIKA DALAM PENJADWALAN MATA KULIAH", *BAREKENG: J. Il. Mat. & Ter.*, vol. 15, no. 04, pp. 615-628, Dec., 2021.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).
Copyright © 2021 Ferdinandus Mone, Justin Eduardo Simarmata

1. PENDAHULUAN

Di era digital ini, perkembangan teknologi mengalami kemajuan yang sangat pesat. Setiap orang mampu mengakses teknologi yang diterapkan di setiap bidang kehidupan. Salah satu yang mengalami kemajuan cukup pesat adalah kemajuan teknologi informasi. Dengan kemajuan teknologi informasi, maka pekerjaan yang awalnya dikerjakan secara manual dan terasa berat menjadi ringan, mampu memberikan informasi yang cepat dan tepat, bahkan dapat mengambil keputusan yang tepat menggunakan teknologi tepat guna [1]. Kemajuan teknologi dan informasi ini sudah banyak dirasakan oleh berbagai instansi baik pemerintah maupun swasta. Teknologi informasi yang digunakan pada instansi perguruan tinggi berhubungan dengan pengolahan data administrasi dan pengambilan keputusan [2].

Dalam pengolahan data administrasi dan pengambilan keputusan pada perguruan tinggi berkaitan dengan pembuatan jadwal mata kuliah. Penjadwalan mata kuliah pada perguruan tinggi khususnya tingkat program studi merupakan suatu hal yang begitu penting dilaksanakan secara rutin pada setiap pergantian semester tahun akademik. Penjadwalan mata kuliah dapat diartikan pengisian kegiatan perkuliahan yang terdiri dari dosen pengampu mata kuliah, ruang kuliah, mata kuliah, mahasiswa, dan jadwal pelaksanaan yang dilaksanakan dalam waktu satu minggu dengan rentang waktu satu hari.

Terdapat beberapa aplikasi penjadwalan yang sudah banyak digunakan seperti *timeline* dan *timetable* namun belum mengakomodir kendala yang dihadapi saat penyusunan jadwal kuliah. Oleh sebab itu, perlu alternatif solusi dalam memecahkan masalah penjadwalan yaitu dengan mengaplikasikan algoritma genetika [3]. Menurut [4] algoritma genetika cukup baik untuk digunakan dalam penjadwalan mata kuliah di sebuah perguruan tinggi. Hal ini sesuai dengan pernyataan [5] bahwa algoritma genetika merupakan salah satu algoritma optimasi yang kuat dan bisa digunakan pada berbagai macam studi kasus seperti penjadwalan serta sering digunakan untuk mencari solusi optimal baik pada kasus yang sederhana sampai yang rumit. Teknik pemecahan masalahnya yaitu menentukan populasi awal dan inialisasi kromosom, menentukan nilai *fitness*, seleksi *crossover*, dan mutasi. Menurut [6] algoritma genetika diterapkan untuk menentukan durasi aktivitas yang optimal sehingga durasi dalam melaksanakan aktivitas tersebut secara keseluruhan dapat diminimalkan dan [7] mengemukakan algoritma genetika dapat menyelesaikan persoalan optimasi untuk mencari perjalanan terpendek. [8] menerapkan algoritma genetika dalam proses optimasi untuk mencari hasil yang terbaik dalam penjadwalan kuliah dengan cara kombinasi perkawinan yang didasari secara acak, mengkodekan dosen, kelas, mata kuliah, ruang, jam, dan hari. Sedangkan menurut [9], algoritma genetika merupakan salah satu cara untuk memecahkan masalah yang cukup besar dengan solusi yang cukup baik meskipun masalah tersebut membutuhkan waktu eksekusi yang lama bila dilakukan secara manual.

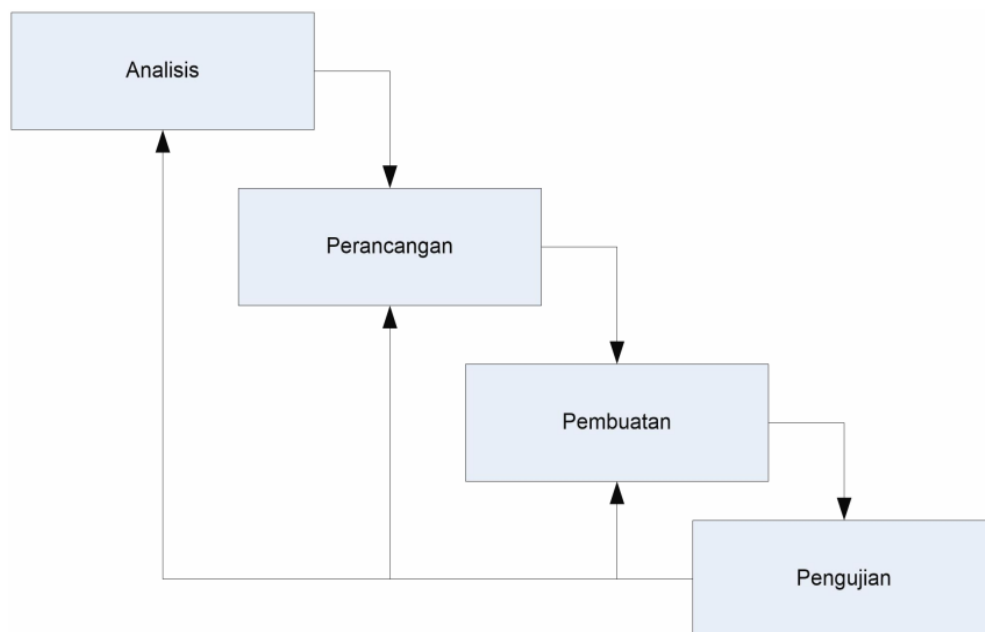
Algoritma genetika atau dikenal dengan *Genetic Algorithm* (GA) adalah teknik pencarian solusi optimum menggunakan prinsip seleksi alam [10]. Algoritma genetika pertama kali dikembangkan oleh John Holland di Universitas Michigan, Amerika Serikat pada tahun 1975, kemudian disebarluaskan oleh muridnya David Golberg pada tahun 1975. Golberg sebagai orang pertama yang mengembangkan GA secara teoritis melalui skema teorema [11]. John Holland menggunakan konsep kromosom untuk menyatakan alternatif solusi dari permasalahan yang diselesaikan menggunakan GA. Kromosom-kromosom terdiri dari beberapa *bit string* yang terdiri dari angka 0 dan 1 yang disebut gen. setiap kromosom kemudian dapat mengalami pertukaran materi genetis yang disebut sebagai perkawinan silang (*cross over*) antar kromosom selanjutnya gen-gen yang terdapat dalam kromosom hasil *cross over* mengalami pergantian secara acak yang disebut sebagai proses mutasi. Selain itu, terdapat proses *inversion* yang mengembalikan urutan beberapa gen yang berurutan di dalam kromosom [12].

Universitas Timor sebagai salah satu perguruan tinggi juga telah merasakan kemajuan teknologi informasi dalam pengolahan data dan pengambilan keputusan, namun terdapat beberapa kendala yang perlu diperbaiki agar sesuai dengan kondisi *real* masalah yang dihadapi. Salah satu kendala yang dihadapi adalah proses pembuatan jadwal mata kuliah di tingkat program studi, khususnya Program Studi Pendidikan Matematika. Program Studi Pendidikan Matematika hanya memiliki 2 ruang kuliah dan 1 ruang aula yang dipakai bersama dengan prodi lain secara bergantian. Selain itu, Program Studi Pendidikan Matematika memiliki 18 dosen yang aktif mengajar pada Semester Genap Tahun Akademik 2020/2021, serta 51 mata kuliah yang disediakan dalam semester tersebut sehingga terdapat 51 pengampu mata kuliah. Untuk mengatasi persoalan yang dialami di Program Studi Pendidikan Matematika mengenai penjadwalan mata kuliah, maka dibuat suatu aplikasi penjadwalan mata kuliah dengan menerapkan algoritma genetika. Algoritma genetika sangat tepat diimplementasikan dalam penjadwalan mata kuliah di perguruan tinggi karena dapat mempersingkat durasi proses pembuatan jadwal serta mampu mengatasi bentrok jadwal

sehingga penjadwalan lebih optimal [15]. Dengan adanya jadwal kuliah yang baik tentunya akan meningkatkan kualitas maupun pelayanan pendidikan. Disamping itu, dengan terbentuknya jadwal kuliah yang baik juga akan berdampak pada penjadwalan sumber daya yang ada, yaitu tenaga pengajar, tenaga kependidikan, serta penggunaan ruang kuliah dengan optimal. Dengan demikian, mahasiswa yang akan mengikuti perkuliahan dengan baik serta optimal [16].

2. METODE PENELITIAN

Dalam pembuatan sistem penjadwalan mata kuliah di Program Studi Pendidikan Matematika Universitas Timor, penulis menggunakan metode waterfall dengan tahapan *Software Development Life Cycle* (SDLC) yang meliputi: analisis, perancangan, pembuatan sistem (*coding*), pengujian (*testing*) [17].



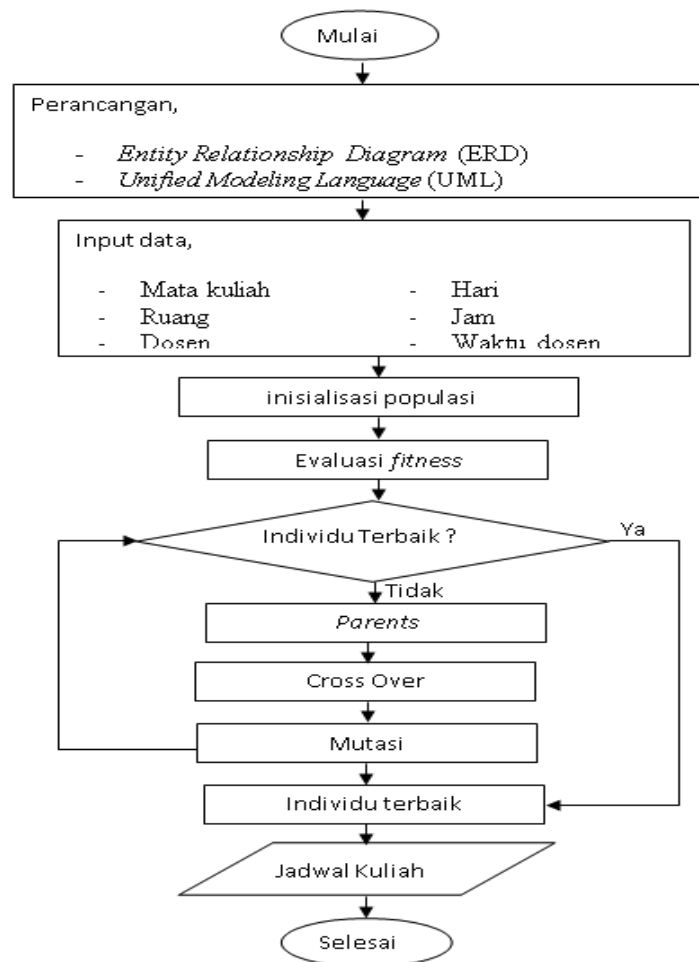
Gambar 1. Tahapan *Software Development Life Cycle*

2.1. Tahap Analisis

Proses analisis masalah merupakan proses untuk memperoleh informasi yang dibutuhkan dalam pembuatan sistem [18]. Tahapan ini dibagi menjadi dua bagian yaitu tahapan pengumpulan data melalui observasi maupun survei tentang masalah *real* yang dihadapi dan tahap mengkaji kendala yang dihadapi selama proses pembuatan jadwal untuk dicarikan solusi dari permasalahan yang dihadapi. Kedua tahapan tersebut kemudian dianalisis untuk mengetahui apakah sistem yang dibuat realistis terhadap masalah waktu, pembiayaan serta lebih efektif dari sistem yang selama ini berjalan [16].

2.2. Perancangan Sistem

Tahap perancangan sistem bertujuan untuk memperoleh gambaran awal berupa langkah-langkah pembuatan sistem aplikasi penjadwalan mata kuliah Program Studi Pendidikan Matematika Universitas Timor. Gambar 2 merupakan diagram alir perancangan aplikasi penjadwalan mata kuliah yang dibuat.



Gambar 2. Diagram Alir Perancangan Aplikasi Penjadwalan Mata Kuliah

2.3. Pembuatan Sistem

Pada tahap ini penulis melakukan *coding system* menggunakan bahasa pemrograman Web yaitu PHP dengan *frame work codeigniter 3*, HTML, CSS yang dikoneksikan dengan database yang dijalankan menggunakan XAMPP 5.6. Proses *coding system* menggunakan code editor Visual Studio Code dengan terlebih dahulu melakukan konfigurasi pada *Code Igniter 3*. Dalam proses *coding* ini juga dirancang *Grappical User Interface (GUI)*. Desain GUI ini merepresentasikan tampilan sistem yang berisi *form input* dan menu yang dibutuhkan pada proses penyusunan jadwal mata kuliah dengan menerapkan Algoritma Genetika. Hasil dari GUI ini akan terlihat sebuah sistem yang telah dibuat.

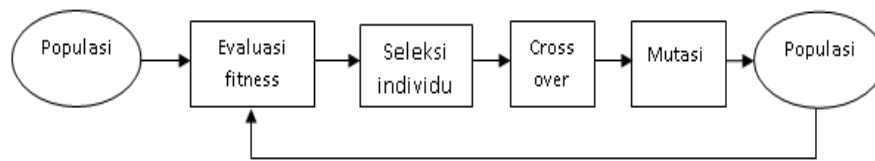
2.4. Pengujian Sistem (Testing)

Testing sistem bertujuan menemukan kesalahan pada sistem dan mencari tahu kesesuaian sistem yang dibuat dengan kebutuhan pengguna. Pada tahap ini penulis menggunakan metode *blackbox testing*.

3. HASIL DAN PEMBAHASAN

Siklus algoritma genetika dimulai dengan membangkitkan populasi secara acak yang merupakan sejumlah calon solusi (individu) kemudian dilanjutkan dengan pengecekan nilai *fitness*, proses perkawinan silang (*cross over*), pengecekan nilai *fitness*, mutasi gen dan diperoleh individu baru. Bila hasil individu baru memiliki nilai *fitness* lebih baik dari proses membangkitkan populasi sebelumnya maka individu baru termasuk salah satu individu dalam populasi baru, jika tidak maka individu yang dibangkitkan sebelumnya kembali diproses lagi dari seleksi *fitness*. Hasil mutasi yang menghasilkan individu baru yang termasuk dalam populasi baru, akan mengeliminasi individu dengan nilai *fitness* yang sangat buruk sehingga jumlah individu dalam populasi baru yang terbentuk sama dengan jumlah individu dalam populasi yang dibangkitkan sejak awal proses. Siklus algoritma ini berhenti apabila nilai *fitness* sudah melewati *hard constraint* maupun *soft constraint* dari masalah yang dihadapi [13]. Selanjutnya menurut [13] algoritma genetika dapat diterapkan

pada beberapa jenis fungsi objektif non-linier dan memiliki fleksibilitas untuk diimplementasikan secara efisien pada masalah tertentu seperti pada gambar 3 berikut.



Gambar 3. Siklus Algoritma Genetika

Penjadwalan merupakan suatu kegiatan alokasi sumber daya dengan memiliki kendala atau batasan yang diberikan kepada suatu objek seperti di ruang-waktu, sedemikian rupa untuk memenuhi sedekat mungkin set tujuan yang diinginkan [9]. Definisi yang lebih umum adalah menugaskan satu set peristiwa (kuliah, kendaraan, acara-acara publik, dll) dengan set terbatas sumber dari waktu ke waktu sedemikian rupa untuk memenuhi *constraint* yang telah ditetapkan. Kendala ini dapat dikategorikan sebagai *hard constraint* dan *soft constraint*, dimana *hard constraint* memiliki prioritas yang lebih tinggi daripada *soft constraint*. Terdapat dua batasan dalam penyusunan penjadwalan kuliah yang dikemukakan [14], yaitu : *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi). *Hard constraint* merupakan batas-batas yang harus diterapkan pada penjadwalan mata kuliah dan harus dipenuhi. Sebuah solusi hanya dapat dikatakan sah dan valid apabila dalam solusi tersebut sama sekali tidak ada *hard constraint* yang terlanggar.

Hard constraint yang umum dalam penjadwalan mata kuliah adalah sebagai berikut:

- Tidak terdapat bentrok dosen terhadap waktu mengajar
- Tidak terdapat bentrok antar dosen pada ruang kuliah tertentu
- Mata kuliah praktikum pada ruang laboratorium

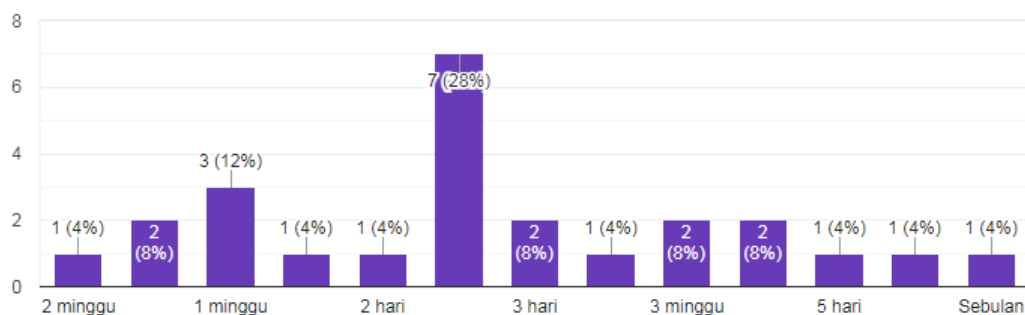
Adapun *soft constraint* yaitu

- Dosen beragama islam tidak mengajar pada jam sholat dzuhur
- Dosen beragama islam tidak mengajar pada jam 11.00 sampai jam 12.30 pada hari jumat

Jika terjadi pelanggaran terhadap kendala yang ditetapkan maka akan diberikan suatu nilai penalti antara 0 sampai 1 untuk setiap pelanggaran. Semakin kecil jumlah pelanggaran yang terjadi, solusi penjadwalan yang dihasilkan akan semakin baik.

3.1. Hasil Analisis Sistem

Dalam proses analisis, penulis menyebarkan angket mengenai waktu yang dibutuhkan dalam pembuatan jadwal dengan menambahkan *soft constraint*. Angket disebar menggunakan *google form* dan diperoleh informasi bahwa 90% tenaga administrasi membutuhkan rata-rata waktu lebih dari 1 hari dalam pembuatan jadwal mata kuliah. Dari angket yang disebar juga diperoleh data bahwa terdapat beberapa cara pegawai tata usaha Fakultas Ilmu Pendidikan Universitas Timor dalam pembuatan jadwal mata kuliah yaitu: 1) menggunakan cara manual/persepsi pribadi, 2) menggunakan aplikasi *time table*, 3) menggunakan teknik pewarnaan sel. Selain itu diperoleh data waktu yang dibutuhkan oleh pegawai tata usaha di Lingkup Fakultas Ilmu Pendidikan Universitas Timor tampak seperti Gambar 4 berikut.

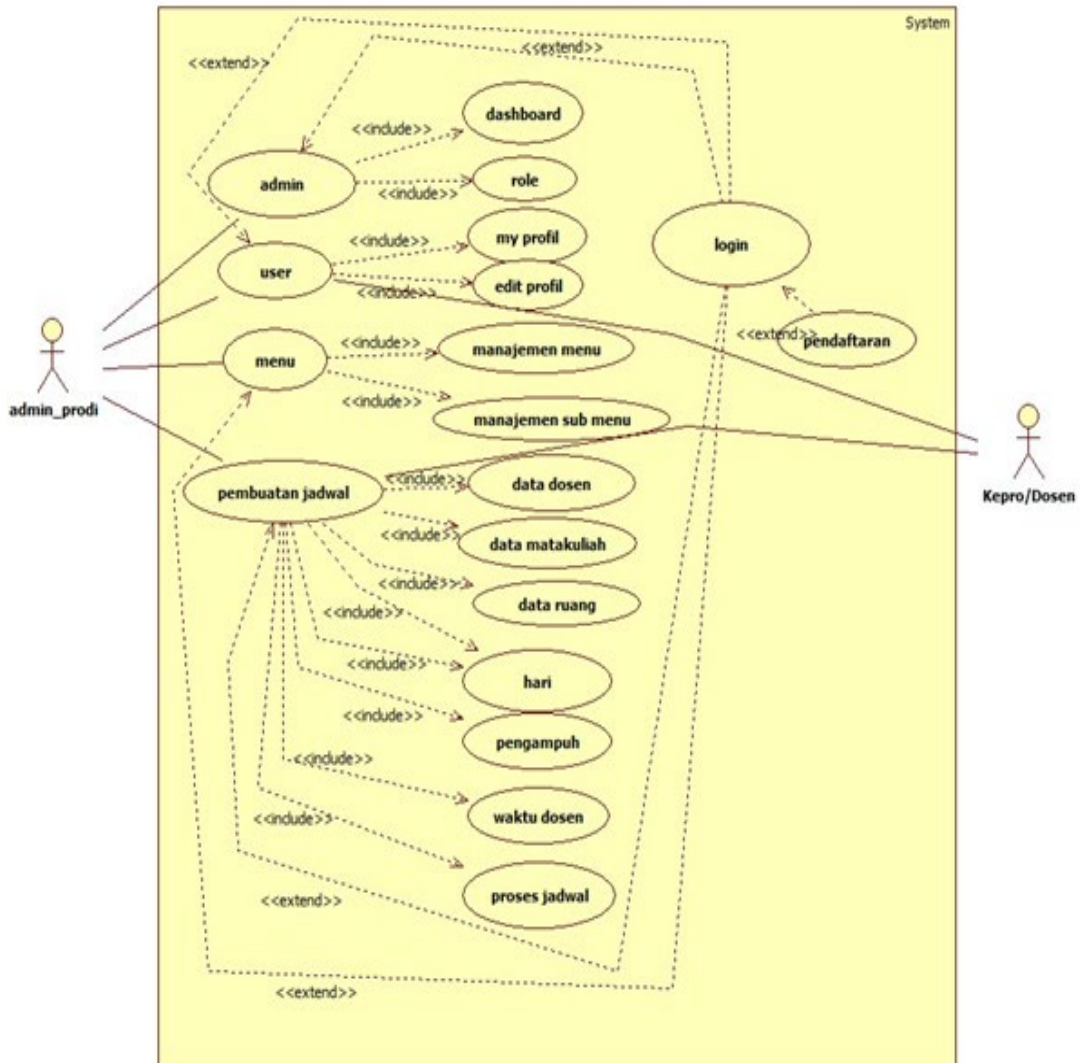


Gambar 4. Waktu Pegawai Tata Usaha Menyusun Jadwal Perkuliahan

Selain kurang efisien waktu pembuatan jadwal, para pegawai TU FIP Universitas Timor merasa sangat membutuhkan sistem penjadwalan yang mampu mengatasi *hard constraint* dan *soft constraint* yang ada.

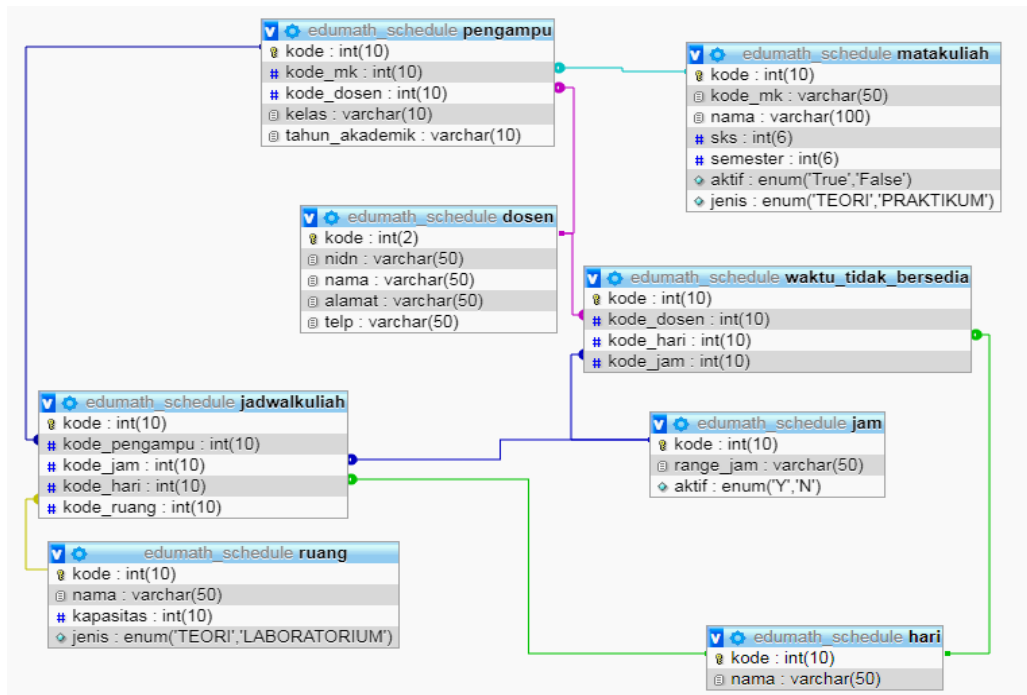
3.2. Hasil Perancangan Sistem

Tahap perancangan merupakan tahap menerjemahkan kebutuhan dalam sebuah representasi perangkat lunak, sebelum dimulai pengimplementasian pada kode program. Penelitian ini menggunakan *Unified Modeling Language* (UML) pada perancangan sistem dan perangkat lunak. UML merupakan sekumpulan diagram yang telah memiliki standar untuk menggambarkan rancangan perangkat lunak berorientasi objek. Diagram UML yang digunakan ialah *use case diagram sequence* dan *class diagram* seperti yang ditunjukkan pada gambar 5 berikut[19].



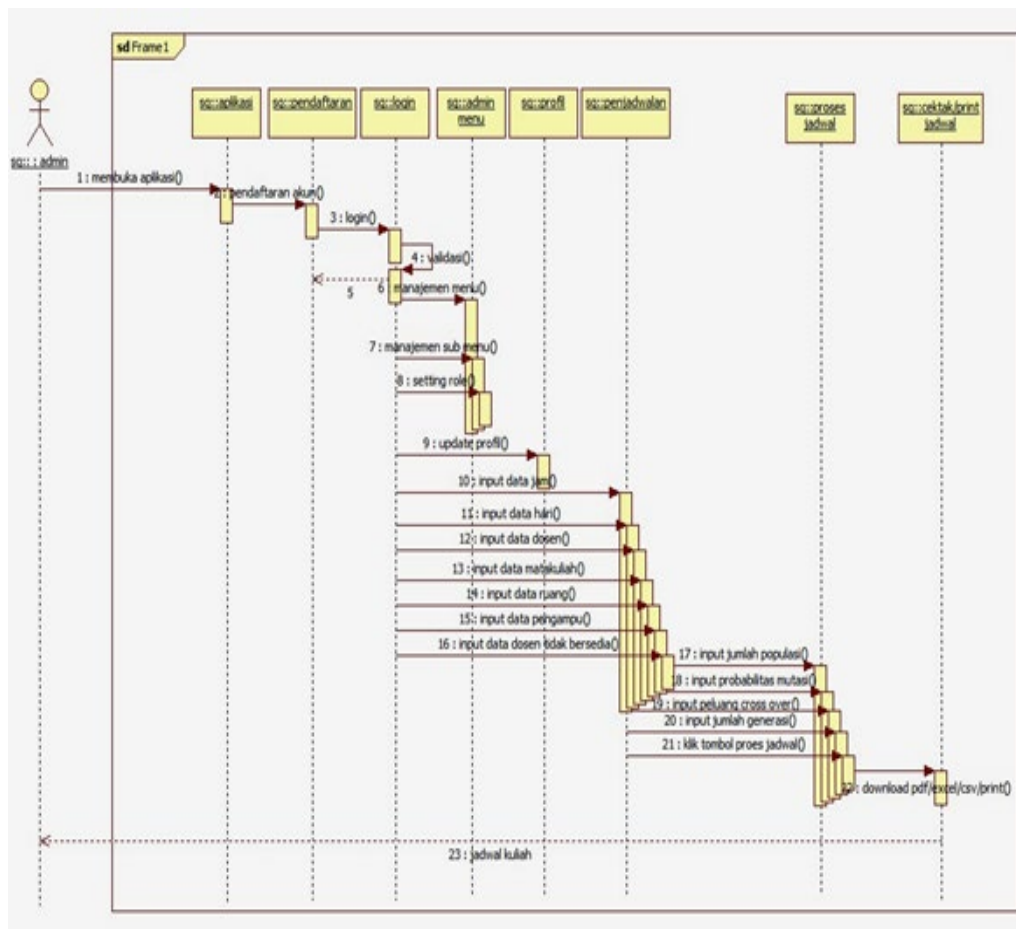
Gambar 5. Unified Modeling Language Diagram

Selain merancang UML diagram dibuat juga *Entity Relationship Diagram* (ERD). ERD digunakan untuk memodelkan model statis suatu sistem. Hasil pembuatan ERD tampak seperti Gambar 6, berikut:



Gambar 6. Hasil Pembuatan Entity Relationship Diagram (ERD)

Perancangan sistem selanjutnya adalah merancang *sequence diagram*. *Sequence diagram* menggambarkan skenario atau langkah – langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu [16]. Hasil perancangan *sequence diagram* tampak seperti Gambar 7, berikut:



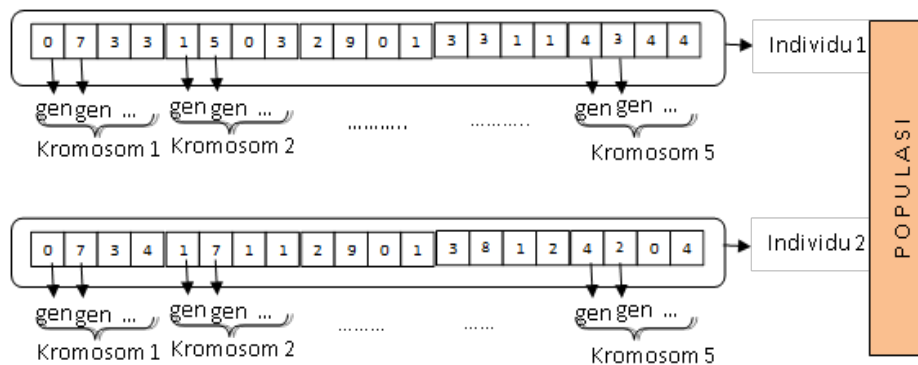
Gambar 7. Hasil Sequence Diagram Aplikasi Mathedu Schedule

3.3. Proses Algoritma Genetika

Algoritma genetika merupakan algoritma yang mengadopsi teori seleksi alam [18]. Algoritma ini dapat mengoptimalkan solusi dari berbagai permasalahan yang dihadapi, salah satunya adalah optimasi pada sistem penjadwalan. Sistem penjadwalan yang dibuat mengikuti diagram alir siklus algoritma genetika yaitu membangkitkan populasi awal, evaluasi *fitness*, seleksi individu, *cross over*, mutasi, dan regenerasi. Proses penerapan algoritma genetika dalam penelitian ini adalah sebagai berikut:

1) Membangkitkan populasi awal

Dalam membangkitkan populasi awal terdapat beberapa *entity* yang kemudian disebut sebagai gen dalam pembentukan kromosom yaitu jam, hari, ruang, pengampu mata kuliah. Kumpulan kromosom ini kemudian membentuk sebuah individu. Dalam penelitian ini, kromosom yang terbentuk sejumlah pengampu mata kuliah. Dalam penelitian ini, satu individu yang terbentuk adalah sebuah jadwal yang dibangkitkan secara random dan belum memenuhi *hard constraint* yang ditetapkan. Contoh populasi yang dibangkitkan dengan jumlah populasi adalah 2 tampak seperti gambar 8 sebagai berikut.



Gambar 8. Contoh Populasi Awal

Sebagai gambaran populasi awal pada individu 1, gen-gen pada kromosom 1 menunjukkan pengampu dengan id = 0 mengajar mata kuliah dengan kode 7 pada kode hari adalah 3 dan kode jam adalah 4. Jadi, kromosom adalah kumpulan gen kode pengampu, gen kode mata kuliah, gen kode hari, dan gen kode jam. Gabungan dari kromosom-kromosom ini adalah individu yang adalah calon jadwal yang terbentuk.

2) Evaluasi *fitness*

Evaluasi *fitness* bertujuan untuk mencari individu terbaik dalam populasi. Individu terbaik ditandai dengan nilai minimum penalti yang diberikan pada individu akibat adanya pelanggaran *hard constraint* dan *soft constraint* antar kromosom. Untuk menghitung nilai *fitness* dari setiap individu digunakan formula

$$f_i = \frac{1}{1 + \sum p_i} \quad (1)$$

dengan f_i adalah *fitness* individu ke- i dan $\sum p_i$ adalah jumlah penalti pada individu ke- i . Setiap pelanggaran dikenakan penalti 1 dan jika tidak terdapat pelanggaran diberi nilai penalti 0.

3) Seleksi individu

Seleksi individu bertujuan untuk memperoleh calon induk yang mampu bertahan hidup. Sebagai contoh jika populasi yang dibangkitkan adalah seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Seleksi Individu

| No | Individu | Penalti | Jumlah Penalti |
|----|--|---|----------------|
| 1 | [0,7,3,3],[1,5,3,3],[2,9,0,1],[3,9,0,1], [4,3,4,4] | Bentrok hari, bentrok jam bentrok dosen | 3 |
| 2 | [0,7,4,3],[1,8,1,1],[2,9,0,1],[3,8,1,2], [4,12,0,4] | Bentrok jam | 1 |
| 3 | [0,7,3,3],[1,10,3,2],[2,1,0,3],[3,2,0,2], [4,2,4,4] | Bentrok dosen, bentrok hari, bentrok jam | 4 |
| 4 | [0,9,1,3],[1,5,3,3],[2,9,3,2],[3,8,2,1], [4,5,1,4] | Bentrok jam karena tidak terdapat matakuliah dengan bobot sks = 1 | 2 |

Maka nilai *fitness* terkecil adalah individu ke-2, kemudian diikuti oleh individu 4. Kedua individu merupakan individu yang akan dipakai pada proses selanjutnya yaitu *cross over*.

4) Cross Over

Proses *cross over* yang dipakai dalam penelitian ini adalah *random cross over*. Proses ini dimulai dengan membangkitkan bilangan bulat acak dari 0 sampai jumlah 1 sebanyak jumlah kromosom. Jika bilangan acak yang dibangkitkan lebih besar dari probabilitas *crossover*, maka kromosom yang bersangkutan mengalami proses pindah silang pada dua individu hasil seleksi. Jika bilangan acak yang dibangkitkan pada kromosom lebih kecil dari probabilitas *cross over*, maka kromosom tersebut tidak terjadi proses *cross over*. Sebagai contoh proses *cross over* pada dua buah individu (*parents*) dengan probabilitas *cross over* 0,7 tampak seperti Tabel 2, berikut.

Tabel 2. Proses Cross Over Pada Dua Buah Individu (*Parents*)

| | Parent 1 | | | | | Parent 2 | | | | |
|-------|-----------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|
| Krom. | [0,7,4,3] | [1,8,1,1] | [2,9,0,1] | [3,8,1,2] | [4,12,0,4] | [0,9,1,3] | [1,5,3,3] | [2,9,3,2] | [3,8,2,1] | [4,5,1,4] |
| acak | 0,5 | 0,1 | 0,8 | 0,9 | 0,7 | 0,5 | 0,1 | 0,8 | 0,9 | 0,7 |
| Ket. | Tidak | Tidak | Cross | Cross | Cross | Tidak | Tidak | Cross | Cross | Cross |

Pada tabel di atas bilangan acak [0, 1] dibangkitkan pada satu individu misalkan *parent 1* dan diperoleh bilangan acak seperti tampak pada tabel (bilangan acak *parent 2* mengikuti bilangan acak *parent 1*). Kromosom yang mengalami *cross over* adalah kromosom 3, kromosom 4, dan kromosom 5, dikarenakan bilangan acak yang dibangkitkan melebihi probabilitas *cross over*. Hasil *cross over* tampak seperti Tabel 3, berikut:

Tabel 3. Kromosom Yang Mengalami Cross Over

| | Child 1 hasil cross over | | | | | Child 2 hasil cross over | | | | |
|-------|--------------------------|-----------|-----------|-----------|-----------|--------------------------|-----------|-----------|-----------|------------|
| Krom. | [0,7,4,3] | [1,8,1,1] | [2,9,3,2] | [3,8,2,1] | [4,5,1,4] | [0,9,1,3] | [1,5,3,3] | [2,9,0,1] | [3,8,1,2] | [4,12,0,4] |

Setelah menghasilkan *child 1* dan *child 2* selanjutnya adalah proses mutasi atau perubahan materi genetik.

5) Mutasi

Proses mutasi adalah proses terjadinya perubahan materi genetik. Dalam penelitian ini materi genetik yang berperan dalam proses mutasi adalah gen hari, gen jam, gen ruang. Proses mutasi dilakukan dengan menetapkan probabilitas mutasi kemudian membangkitkan bilangan acak diantara 0 dan 1. Jika bilangan acak yang dibangkitkan melebihi probabilitas mutasi, maka gen yang bersangkutan mengalami proses mutasi. Sebagai contoh, bila di set probabilitas mutasi adalah 0,4 dan bilangan acak yang dibangkitkan tampak seperti pada Tabel 4, maka kromosom yang akan mengalami mutasi sebagai berikut.

Tabel 4. Kromosom Yang Akan Mengalami Mutasi

| | Mutan 1 sebelum mutasi | | | | | Mutan 2 sebelum mutasi | | | | |
|-------|------------------------|-----------|-----------|-----------|-----------|------------------------|-----------|-----------|-----------|------------|
| Krom. | [0,7,4,3] | [1,8,1,1] | [2,9,3,2] | [3,8,2,1] | [4,5,1,4] | [0,9,1,3] | [1,5,3,3] | [2,9,0,1] | [3,8,1,2] | [4,12,0,4] |
| Acak | 0,4 | 0,1 | 0,2 | 0,5 | 0,9 | 0,1 | 0,5 | 0,1 | 0,4 | 0,5 |
| Ket | mutasi | tidak | Tidak | mutasi | mutasi | tidak | mutasi | tidak | mutasi | mutasi |

Pada mutan 1: kromosom 1, kromosom 4, dan kromosom 5, mengalami mutasi sedangkan pada mutan 2: kromosom 2, kromosom 4, dan kromosom 5. Jadi materi genetik hari dan jam mengalami mutasi pada kromosom yang memenuhi syarat mutasi. Contoh hasil mutasi tampak seperti Tabel 5, berikut:

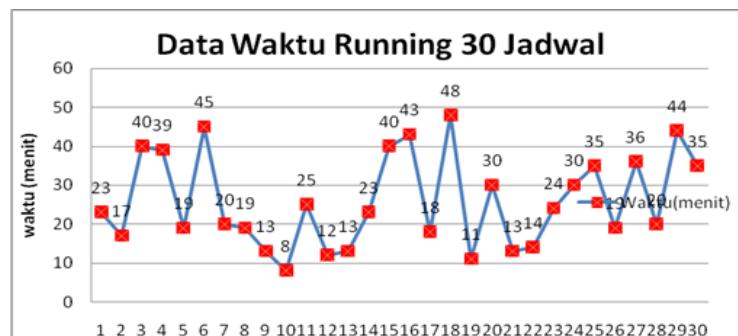
Tabel 5. Hasil Mutasi

| | Mutan 1 sebelum mutasi | | | | | Mutan 2 sebelum mutasi | | | | |
|-------|------------------------|-----------|-----------|-----------|-----------|------------------------|-----------|-----------|-----------|------------|
| Krom. | [0,7,3,3] | [1,8,1,1] | [2,9,3,2] | [3,8,4,1] | [4,5,5,2] | [0,9,1,3] | [1,5,5,1] | [2,9,0,1] | [3,8,1,7] | [4,12,1,4] |

Proses mutasi adalah akhir dari siklus algoritma genetika pada sebuah generasi. Jika hasil mutasi ini dievaluasi *fitness* nya dan *fitness* nya dikategorikan baik, maka mutan 1 atau mutan 2 masuk dalam populasi yang menjadi proses untuk siklus pada generasi 2. Siklus algoritma genetika ini akan dilanjutkan sampai ditemukan individu terbaik dalam hal ini diperoleh jadwal yang memenuhi *hard constraint* dan *soft constraint*.

3.4. Hasil Pengujian Sistem (Testing)

Testing sistem bertujuan menemukan kesalahan pada sistem dan mencari tahu kesesuaian sistem yang dibuat dengan kebutuhan pengguna. Pada tahap ini penulis menggunakan metode *blackbox testing*. Pengujian sistem penjadwalan yang telah dibuat dijalankan menggunakan *Server Local Apache*, menggunakan laptop dengan spesifikasi RAM 2.0 GB. Data hasil *running* aplikasi mengukur waktu eksekusi penjadwalan yang telah dibuat menggunakan algoritma genetika. Hasil *running* aplikasi diperoleh rata-rata waktu eksekusi 30 jadwal adalah 25.86 menit, standar deviasi 11,88 menit dengan jumlah ruang kuliah sebanyak 3 ruang kuliah, dan 1 ruang aula untuk mata kuliah umum, 51 pengampu mata kuliah, 18 dosen, 5 hari kerja dan 14 jam efektif per hari. Hasil *running* tampak seperti Gambar 9, berikut:



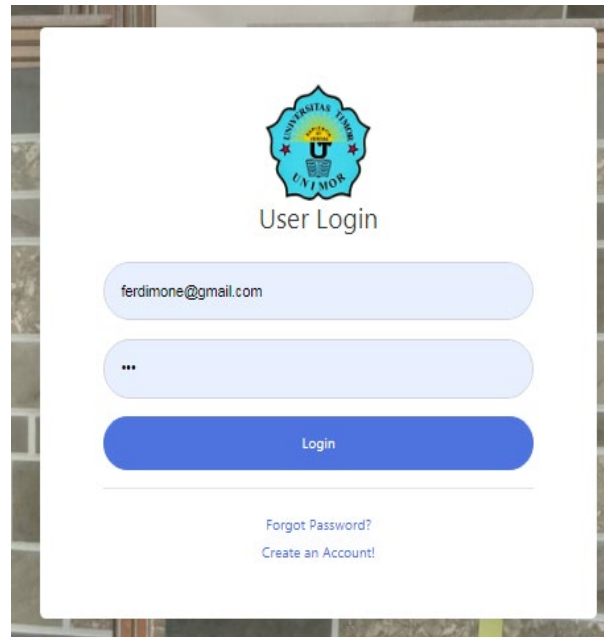
Gambar 9. Data Waktu Running 30 Jadwal

Pengujian proses algoritma genetika dalam proses pembuatan aplikasi penjadwalan mata kuliah Program Studi Pendidikan Matematika Unimor tampak seperti Tabel 6, berikut:

| Tabel 6. Hasil Pengujian Proses Algoritma Genetika pada Aplikasi Penjadwalan Mata kuliah | | |
|--|---|--------------------------|
| Jenis Constraint | Constraint | Hasil Pengujian (Respon) |
| Hard Constraint | ▪ Tidak terdapat bentrok dosen terhadap waktu mengajar | Sukses |
| | ▪ Tidak terdapat bentrok antar dosen pada ruang kuliah tertentu | Sukses |
| | ▪ Mata kuliah praktikum pada ruang laboratorium | Sukses |
| Soft Constraint | ▪ Dosen beragama islam tidak mengajar pada jam sholat dzuhur | Sukses |
| | ▪ Dosen beragama islam tidak mengajar pada jam 11.00 sampai jam 12.30 pada hari jumat | Sukses |

Hasil pengujian di atas menunjukkan bahwa aplikasi penjadwalan sukses dijalankan menggunakan perangkat berupa laptop dengan RAM 2.0 Gb sehingga aplikasi penjadwalan siap diimplementasikan di tingkat program Studi Pendidikan Matematika Universitas Timor. Selain pengujian proses algoritma genetika di atas,

pengujian terhadap *event* dan *response* pada GUI juga menunjukkan sukses (tidak terjadi *error*) dalam GUI. Hasil GUI ini menampilkan secara fisik aplikasi penjadwalan yang dinamakan *Edumath-Schedule*. Aplikasi ini terdiri dari dua jenis aktor yaitu untuk admin dan untuk *user* (Dosen/Ketua Program Studi). Tampilan halaman *login* tampak seperti pada Gambar 10.



Gambar 10. Halaman Login Edumath-Schedule

Pada halaman *login user* yang belum mempunyai akun harus mendaftarkan terlebih dahulu dan memperoleh *email* verifikasi dan *user* membuka *email* untuk verifikasi. Jika terjadi lupa *password* maka *user* dapat menggunakan menu lupa *password* kemudian menerima token pada *email* untuk mereset *password*. Setelah dilakukan pendaftaran akun dan berhasil, maka *user* diarahkan pada halaman awal aplikasi, seperti yang ditunjukkan pada Gambar 11, berikut:



Gambar 11. Halaman Utama Tampilan Admin

Gambar 11, di atas adalah bagian beranda dari tampilan aplikasi (halaman utama). Pada bagian ini, terdapat semua menu untuk mengoperasikan aplikasi ini. Selain itu, telah dimuat video tutorial bagaimana penggunaan

aplikasi mulai dari langkah registrasi akun, input master data sampai pada proses pembuatan jadwal. Data yang diperlukan untuk proses pembuatan jadwal kuliah adalah data dosen, data ruang, data mata kuliah, data hari, data jam, data pengampu, dan data waktu dosen tidak bersedia. Proses *input* data pada data master ini disediakan menu *import* dengan terlebih dahulu mengunduh *format import* atau bisa di *input* melalui *form* dengan menekan tombol, misalkan tambah dosen. Selain itu, disediakan juga menu edit dan hapus data pada setiap baris data. Hasil *input* data dapat diunduh dalam format csv, xlsx, atau pdf. Halaman *input* data dosen ditunjukkan seperti pada Gambar 12 berikut:

| # | NIDN | Nama | Telp | Aksi |
|----|-------------|---------------------------------------|--------------|----------------|
| 01 | 08300577901 | Alosius Loka Son, s.Pd.,M.Pd | 081326666666 | [Edit] [Hapus] |
| 02 | 0000002 | Cecilia Novianti Salsinha, S.Si,M.Sc | 081000000002 | [Edit] [Hapus] |
| 03 | 0810000 | Dominifridus Bone | 086554421 | [Edit] [Hapus] |
| 04 | 0000008 | E. Kristanti, S.Psi,MA | 081000000008 | [Edit] [Hapus] |
| 05 | 0000020 | Eduardus Beo Sesu Delvion, S.Si,M.Mat | 081000000020 | [Edit] [Hapus] |
| 06 | 0000xxxxxx | Emanuel Maria Yosefh Hano'e,M.Ling | 08xyyy | [Edit] [Hapus] |
| 07 | 00111122233 | Emanuel Ulu | 08887766611 | [Edit] [Hapus] |
| 08 | 830058802 | Ferdinandus Mone, S.Pd.,M.Pd | 081000000001 | [Edit] [Hapus] |
| 09 | 00xxxxxx | Ferdinandus Siki, M.Pd | 08xxxx | [Edit] [Hapus] |

Gambar 12. Master Data Dosen

Proses pembuatan jadwal dapat dijalankan jika semua data yang diperlukan untuk pembuatan jadwal telah berhasil di *input*. Tampilan halaman proses jadwal tampak seperti Gambar 13.

| No | Hari | Sesi | Jam | Matakuliah | SKS | Semester | Kelas | Dosen | Ruang |
|----|-------|---------|-------------|--------------------|-----|----------|-------|---------------------------------------|-------|
| 1 | Senin | (1-3) | 07.00-09.30 | Analisa Numerik | 3 | 5 | A | Meiva Marthaulina Lestari, S.Pd.,M.Sc | FIP 7 |
| 2 | Senin | (4-6) | 09.30-12.00 | statistika Lanjut | 3 | 3 | B | Selestina Nahak, S.Pd.,M.Pd | FIP 7 |
| 3 | Senin | (8-10) | 12.50-15.20 | Kalkulus II | 3 | 3 | B | Zulkaidah Nur Ahzan, S.Si,M.Si | FIP 7 |
| 4 | Senin | (11-13) | 15.20-17.50 | Aplikasi Komputer | 3 | 3 | A | Ferdinandus Mone, S.Pd.,M.Pd | FIP 6 |
| 5 | Rabu | (8-9) | 12.50-14.30 | Media Pembelajaran | 2 | 3 | A | Talisadika S. Marfa, S.Pd.,M.Pd | FIP 6 |
| 6 | Rabu | (10-12) | 14.30-17.00 | Kalkulus II | 3 | 3 | C | Ferdinandus Mone, S.Pd.,M.Pd | AKBID |

Gambar 13. Tampilan Proses Jadwal

Gambar 13, di atas adalah tampilan menu proses jadwal. Jadwal diproses menggunakan alur algoritma genetika, sehingga perlu melakukan *input* parameter yang diperlukan seperti: tahun akademik, semester, jumlah populasi, probabilitas *cross over*, probabilitas mutasi, dan jumlah generasi. Secara *default*, peneliti memberikan probabilitas *cross over* 0.7, probabilitas mutasi 0.3, jumlah populasi 10, dan jumlah generasi 1000. Hasil *running* jadwal dapat diunduh dalam format excel, csv ataupun pdf. Hasil dalam bentuk pdf tampak seperti Gambar 14.

| No | Hari | Sesi | Jam | Matakuliah | SKS | Semester | Kelas | Dosen | Ruang |
|----|--------|---------|-------------|----------------------------|-----|----------|-------|--|----------|
| 1 | Senin | (2-3) | 07.50-09.30 | Micro Teaching | 2 | 6 | B | Hermina Disnawati, S.Pd.,M.Pd | FIP 7 |
| 2 | Senin | (2-4) | 07.50-10.20 | Analisa Kompleks | 3 | 6 | B | Fitriani, S.Si.,M.Sc | FIP 6 |
| 3 | Senin | (4-5) | 09.30-11.10 | Bahasa Indonesia | 2 | 2 | A | Joni S. Nalanan, S.Pd., M.Hum | Aula FIP |
| 4 | Senin | (4-6) | 09.30-12.00 | Analisis Vektor | 3 | 6 | B | Meiva Marthaulina Lestari, S.Pd.,M.Sc | FIP 7 |
| 5 | Senin | (6-8) | 11.10-13.40 | Analisis Vektor | 3 | 6 | A | Yohanis Ndapa Deda, S.Pd.,M.Si | FIP 6 |
| 6 | Senin | (7-9) | 12.00-14.30 | Geometri Analitika | 3 | 2 | B | Hendrika Bete, S.Pd.,M.Pd | FIP 7 |
| 7 | Senin | (9-11) | 13.40-16.10 | Statistika Dasar | 3 | 2 | A | Cecilia Novianti Salsinha, S.Si.,M.Sc | FIP 6 |
| 8 | Senin | (10-11) | 14.30-16.10 | Micro Teaching | 2 | 6 | A | Talisadika S. Maifa, S.Pd.,M.Pd | FIP 7 |
| 9 | Senin | (12-14) | 16.10-18.40 | Operasi Riset | 3 | 6 | A | Eduardus Beo Sesu Delvion, S.Si.,M.Mat | FIP 6 |
| 10 | Senin | (12-14) | 16.10-18.40 | Kapita Selektta Matematika | 3 | 6 | B | Hermina Disnawati, S.Pd.,M.Pd | FIP 7 |
| 11 | Selasa | (1-2) | 07.00-08.40 | Teori Bilangan | 2 | 2 | A | Zulkaidah Nur Ahzan, S.Si.,M.Si | FIP 6 |

Gambar 14. Jadwal Kuliah Hasil Download Format Pdf

4. KESIMPULAN

Hasil *running* aplikasi diperoleh rata-rata waktu eksekusi 30 jadwal adalah 25.86 menit, standar deviasi 11,88 menit dengan jumlah ruang kuliah sebanyak 3 ruang kuliah, dan 1 ruang aula untuk mata kuliah umum, 51 pengampu mata kuliah, 18 dosen, 5 hari kerja dan 14 jam efektif per hari. Selain dari hasil tersebut, diperoleh bahwa dari 30 jadwal yang diproses secara berturut-turut memberikan kesimpulan yakni, (1) tidak terjadi bentrok dosen, (2) tidak terjadi bentrok ruang dan waktu, (3) tidak terjadi bentrok dengan waktu dosen yang berhalangan, (4) tidak terjadi bentrok dengan waktu sholat jumat, (5) ruang Aula FIP digunakan untuk mata kuliah yang sudah ditetapkan sebelum proses jadwal. Berdasarkan hasil pengujian sistem disimpulkan bahwa aplikasi penjadwalan yang dibuat efektif dan efisien.

UCAPAN TERIMA KASIH

Terima kasih kepada Ketua UPT LPPM Universitas Timor yang telah memberikan dukungan melalui surat penugasan No.010/UN60/LPPM/PP/2021 serta seluruh rekan yang turut dalam proses penelitian ini.

DAFTAR PUSTAKA

- [1] Z. Zhou, F. Li, H. Zhu, H. Xie, J. H. Abawajy, and M. U. Chowdhury, "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1531–1541, Mar. 2020, doi: 10.1007/s00521-019-04119-7.
- [2] A. Josi, "Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall (Studi Kasus: STMIK Prabumulih)," *J. Inform. Pengemb. IT*, vol. 02, no. 02, pp. 77–83, 2017, Accessed: Apr. 01, 2021. [Online]. Available: <http://ejournal.poltektegal.ac.id/index.php/informatika/article/view/517>.
- [3] Y. Sari, M. Alkaff, E. S. Wijaya, S. Soraya, and D. P. Kartikasari, "Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika Dengan Teknik Tournament Selection," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 1, pp. 85–92, 2019, doi: 10.25126/jtiik.201961262.
- [4] L. Tambunan, "Implementasi Algoritma Genetika dalam Pembuatan Jadwal Kuliah," *Jar. Sist. Inf. Robot.*, vol. 1, no. 01, pp. 1–7, 2017.
- [5] Y. Elva, "Sistem Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika," *J. Teknol. Inf.*, vol. 3, no. 1, pp. 49–57, 2019, Accessed: Apr. 03, 2021. [Online]. Available: <http://jurnal.una.ac.id/index.php/jurti/article/view/687>.
- [6] L. D. Long and A. Ohsato, "Solving the resource-constrained project scheduling problem by genetic algorithm," *J. Japan Ind. Manag. Assoc.*, vol. 57, no. 6, pp. 520–529, 2007, doi: 10.1061/(asce)cp.1943-5487.0000874.
- [7] J. E. Simarmata, "Penerapan Algoritma Branch And Bound Pada Persoalan Pedagang Keliling (Travelling Salesman Problem),"

- RANGE J. Pendidik. Mat.*, vol. 1, no. 2, pp. 111–121, 2020, doi: <https://doi.org/10.32938/jpm.v1i2.366>.
- [8] A. T. Ma'arif and D. P. Pamungkas, "Penerapan Metode Algoritma Genetika Untuk Optimasi Penjadwalan Mata Kuliah," *Pros. SEMNAS INOTEK (Seminar Nas. Inov. Teknol.)*, vol. 4, no. 2, pp. 93–97, 2020, doi: <https://doi.org/10.29407/inotek.v4i2.126>.
- [9] X. Zan, Z. Wu, C. Guo, and Z. Yu, "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems," *Adv. Mech. Eng.*, vol. 12, no. 1, pp. 1–15, 2020, doi: [10.1177/1687814019885294](https://doi.org/10.1177/1687814019885294).
- [10] M. Fera, F. Fruggiero, A. Lambiase, R. Macchiaroli, and V. Todisco, "A modified genetic algorithm for time and cost optimization of an additive manufacturing single-machine scheduling," *Int. J. Ind. Eng. Comput.*, vol. 9, no. 4, pp. 423–438, 2018, doi: [10.5267/j.ijiec.2018.1.001](https://doi.org/10.5267/j.ijiec.2018.1.001).
- [11] K. Peng et al., "A Hybrid Genetic Algorithm on Routing and Scheduling for Vehicle-Assisted Multi-Drone Parcel Delivery," *IEEE Access*, vol. 7, pp. 49191–49200, 2019, doi: [10.1109/ACCESS.2019.2910134](https://doi.org/10.1109/ACCESS.2019.2910134).
- [12] A. Pramana Henriyan, B. Hananto, F. Muhammad Ardi, H. Lenggana, P. Ulum, and F. Desti, "Implementasi Algoritma Genetika pada Sistem Informasi Penjadwalan Matakuliah," in *SEMNASATI*, 2019, pp. 283–287, Accessed: Apr. 01, 2021. [Online]. Available: <http://prosiding.uika-bogor.ac.id/index.php/semnati/article/view/307>.
- [13] Ferdyawan and A. Hajjah, "Penerapan Algoritma Genetika dalam Optimasi Penjadwalan Proyek," *J. Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 2, no. 1, pp. 50–55, 2020, Accessed: Apr. 02, 2021. [Online]. Available: <http://www.ejournal.pelitaIndonesia.ac.id/JMAPTeKsi/index.php/JOM/article/view/545>.
- [14] M. Sanapiah, A. Rofi'ah, H. Jayanti, Alysha Ghea Arliana, and V. N. Wijyaningrum, "Penyusunan Jadwal Asisten Praktikum Menggunakan Algoritma Genetika," *Sist. J. Sist. Inf.*, vol. 9, no. 1, pp. 282–287, 2020, doi: <https://doi.org/10.32520/stmsi.v8i2.501>.
- [15] R. Christian and D. S. Donoriyanto, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Kuliah Program Studi Teknik Industri Upn "Veteran" Jawa Timur," *Tekmapro J. Ind. Eng. Manag.*, vol. 16, no. 2, pp. 1–12, 2021, doi: [10.33005/tekmapro.v16i2.157](https://doi.org/10.33005/tekmapro.v16i2.157).
- [16] Y. S. Dwanoko, "Implementasi Software Development Life Cycle (Sdlc) Dalam Penerapan Pembangunan Aplikasi Perangkat," *J. Teknol. Inf.*, vol. 7, no. 2, pp. 83–94, 2016.
- [17] D. Asmarajati, M. Fuat Asnawi, and R. D. Akmal, "Implementasi Algoritma Genetika Pada Penjadwalan Sistem Informasi XYZ TV," *Riau J. Comput. Sci.*, vol. 06, no. 01, pp. 12–22, 2020.
- [18] M. Susanti, S. Nusa, and M. Jakarta, "Perancangan Sistem Informasi Akademik Berbasis Web Pada Smk Pasar Minggu Jakarta," *J. Inform.*, vol. III, no. 1, pp. 57–64, 2016, Accessed: Apr. 02, 2021. [Online]. Available: <http://ijns.org/journal/index.php/ijns/article/view/288>.
- [19] Rosmila, M. Yamin, and L. Tajidun, "Aplikasi Pembagian Harta Waris Menurut Hukum Islam Dengan Menggunakan Metode Algoritma Genetika," *ojs.uho.ac.id*, vol. 2, no. 2, pp. 225–236, 2016, Accessed: Apr. 03, 2021. [Online]. Available: <http://ojs.uho.ac.id/index.php/semantik/article/view/1849>.