

SKRIPSI

**OTOMATISASI SISTEM PENJADWALAN UJIAN
AKHIR SEMESTER MENGGUNAKAN METODE
ALGORITMA GENETIKA**

**(Studi Kasus : Fakultas Ilmu Kesehatan Universitas Muhammadiyah
Magelang)**



AGUNG DWI NOFIANTO

NPM : 14.0504.0045

**PROGRAM STUDI TEKNIK INFORMATIKA S1
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAGELANG
TAHUN 2020**

SKRIPSI

**OTOMATISASI SISTEM PENJADWALAN UJIAN
AKHIR SEMESTER MENGGUNAKAN METODE
ALGORITMA GENETIKA**

**(Studi Kasus : Fakultas Ilmu Kesehatan Universitas Muhammadiyah
Magelang)**



AGUNG DWI NOFIANTO

NPM : 14.0504.0045

**PROGRAM STUDI TEKNIK INFORMATIKA S1
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAGELANG
TAHUN 2020**

SKRIPSI

**OTOMATISASI SISTEM PENJADWALAN UJIAN
AKHIR SEMESTER MENGGUNAKAN METODE
ALGORITMA GENETIKA**

**(Studi Kasus : Fakultas Ilmu Kesehatan Universitas Muhammadiyah
Magelang)**

Disusun Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana Komputer
(S.Kom) Program Studi Teknik Informatika Jenjang Strata Satu (S-1) Fakultas
Teknik Universitas Muhammadiyah Magelang



AGUNG DWI NOFIANTO
NPM. 14.0504.0045

PROGRAM STUDI TEKNIK INFORMATIKA S1
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAGELANG
TAHUN 2020


HALAMAN PENEGASAN

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Agung Dwi Nofianto

NPM : 14.0504.0045

Magelang, 14 Februari 2020



AGUNG DWI NOFIANTO
NPM. 14.0504.0045

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Agung Dwi Nofianto

NPM : 14.0504.0045

Program Studi : Teknik Informatika S1

Fakultas : Teknik

Alamat : Bayem RT/01 RW/04 Kutoarjo, Purworejo

Judul Skripsi : OTOMATISASI SISTEM PENJADWALAN UJIAN AKHIR
SEMESTER MUNGGUNAKAN ALGORITMA GENETIKA
(Studi Kasus : Fakultas Ilmu Kesehatan Universitas
Muhammadiyah Magelang)

Dengan ini menyatakan bahwa Skripsi ini merupakan hasil karya sendiri dan bukan merupakan plagiat dari hasil karya orang lain. Dan bila di kemudian hari terbukti bahwa karya ini merupakan plagiat, maka saya bersedia menerima sanksi administrasi maupun sanksi apapun.

Demikian surat pernyataan ini saya buat dengan penuh kesadaran dan sebenarnya serta penuh tanggung jawab.

Magelang, 14 Februari 2020

Yang menyatakan,



AGUNG DWI NOFIANTO

NPM. 14.0504.0045

KATA PENGANTAR


Puji syukur kehadiran Allah SWT, atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer di Program Studi Teknik Informatika S1 Fakultas Teknik Universitas Muhammadiyah Magelang.

Penyelesaian Skripsi ini banyak memperoleh bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, diucapkan terima kasih sebesar – besarnya kepada :

1. Allah SWT yang telah memudahkan segala urusan saya, sehingga dapat menyelesaikan skripsi ini.
2. Kedua orang tua dan keluarga yang telah memberikan dukungan baik secara moril maupun materi hingga terselesaikannya skripsi ini.
3. Yun Arifatul Fatimah, S.T., M.T., Ph.D. selaku Dekan Fakultas Teknik Universitas Muhammadiyah Magelang.
4. Agus Setiawan, M.Eng. selaku Ketua Program Studi Teknik Informatika S1 Universitas Muhammadiyah Magelang.
5. Nugroho Agung Prabowo, S.T., M.Kom dan Emilya Uly Artha, M.Kom selaku Dosen pembimbing, pendamping yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penyusunan skripsi ini.
6. Seluruh Dosen Fakultas Teknik Universitas Muhammadiyah Magelang yang telah memberikan ilmu dan pengetahuan yang bermanfaat.
7. Keluarga besar Kos Rustam, Mas Pras, dan Bu Tika Sekeluarga, yang selalu memberikan semangat dan doanya.

Akhir kata, semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu yang tidak dapat disebutkan namanya satu persatu dan semoga Skripsi ini membawa manfaat bagi semua pihak.

Magelang, 14 Februari 2020


AGUNG DWI NOFIANTO
NPM. 14.0504.0045

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Muhammadiyah Magelang, yang bertandatangan di bawah ini :

Nama : Agung Dwi Nofianto
NPM : 14.0504.0045
Program Studi : Teknik Informatika S1
Fakultas : Teknik
Jenis karya : Skripsi

Menyatakan bahwa demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Fakultas Teknik Universitas Muhammadiyah Magelang **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah yang berjudul :


Otomatisasi Sistem Penjadwalan Ujian Akhir Semester Menggunakan Metode Algoritma Genetika (Studi Kasus : Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Fakultas Teknik Universitas Muhammadiyah Magelang berhak menyimpan, mengalihmedia/memformatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan Skripsi tersebut selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini dibuat dengan sebenar-benarnya tanpa paksaan dari pihak manapun.

Dibuat di : Magelang

Pada tanggal : 14 Februari 2020

Yang menyatakan


AGUNG DWI NOFIANTO
NPM. 14.0504.0045

ABSTRAK

OTOMATISASI SISTEM PENJADWALAN UJIAN AKHIR SEMESTER MENGGUNAKAN METODE ALGORITMA GENETIKA (STUDI KASUS : FAKULTAS ILMU KESEHATAN UNIVERSITAS MUHAMMADIYAH MAGELANG)

Nama : Agung Dwi Nofianto

Pembimbing : 1. Nugroho Agung Prabowo, S.T., M.Kom
2. Emilya Ully Artha, M.Kom

Permasalahan pembuatan penjadwalan ujian merupakan permasalahan yang membutuhkan banyak waktu dalam proses penyelesaiannya dan masalah ini sering dihadapi oleh Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang dikarenakan proses penjadwalan ujian akhir semester masih dilakukan secara manual menggunakan *microsoft excel*. Tujuan dari penelitian adalah menemukan cara menjadwalkan sejumlah komponen yang terdiri dari mata kuliah, ruang, waktu dan pengawas dengan batasan-batasan tertentu. Untuk mempermudah proses otomatisasi penjadwalan penulis menggunakan metode Algoritma Genetika, yaitu Algoritma yang menggunakan prinsip genetika dan seleksi alamiah dalam proses optimisasi penjadwalan. Algoritma ini dimulai dengan beberapa kombinasi populasi, yang terdiri dari kromosom-kromosom yang berisi data mata kuliah, waktu, ruangan dan pengawas, dan dilakukan proses seleksi untuk mendapatkan solusi terbaik. Hasil yang dicapai dari penelitian ini berupa jadwal yang optimum dimana jadwal yang diolah dengan jumlah kromosom 50 dan maksimal generasi 100 menghasilkan nilai *fitness* 1.

Kata kunci : *Penjadwalan, Algoritma genetika , Otomatis*

ABSTRACT

AUTOMATION OF SEMESTER FINAL EXAM SCHEDULING SYSTEM USING THE GENETIK ALGORITHM METHOD (CASE STUDY : FACULTY OF HEALTH SCIENCES, MUHAMMADIYAH UNIVERSITY, MAGELANG)

Nama : Agung Dwi Nofianto

Pembimbing : 1. Nugroho Agung Prabowo, S.T., M.Kom

2. Emilya Uilly Artha, M.Kom

The problem of making exam scheduling is a problem that requires a lot of time in the process, and this problem is often faced by the Faculty of Health Sciences, University of Muhammadiyah Magelang since the final semester exam scheduling process is still done manually using Microsoft Excel. The purpose of the research is to find a way the scheduling to describe all components consisting of courses, classroom, time and supervisors with certain limitations. To facilitate the scheduling automation process the author used the Genetic Algorithm method, which is an algorithm that uses the principles of genetics and natural selection in the process of optimizing scheduling. This algorithm starts with various combinations, which consist of chromosomes containing course data, time, classroom and supervisor, and the selection process is carried out to get the best solution. The results obtained from this study consist of an optimal schedule in which the schedule is processed with the number of chromosomes of 50 and the maximum generation of 100 produces a fitness value of 1.

Keywords: *Scheduling, Genetic Algorithms, Automation*

DAFTAR ISI

HALAMAN KULIT MUKA	ii
HALAMAN JUDUL.....	iii
HALAMAN PENEGASAN.....	iv
PERNYATAAN KEASLIAN.....	v
HALAMAN PENGESAHAN.....	vi
KATA PENGANTAR	vii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	viii
ABSTRAK	ix
ABSTRACT	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB 1	1
A. LATAR BELAKANG	1
B. PERUMUSAN MASALAH	3
C. BATASAN MASALAH	3
D. TUJUAN PENELITIAN	4
E. MANFAAT PENELITIAN.....	4
BAB II.....	5
A. PENELITIAN RELEVAN.....	5
B. PENJELASAN TEORITIS MASING – MASING VARIABEL PENELITIAN.....	7
C. LANDASAN TEORI.....	20
BAB III	21
A. METODE PENELITIAN.....	21
B. ANALISIS SISTEM	31
C. PERANCANGAN UNIFIED MODELING LANGUAGE DIAGRAM...	46
D. PERANCANGAN BASIS DATA	50
E. DESAIN DATABASE.....	55
F. PERANCANGAN INTERFACE.....	58
BAB IV	63
IMPLEMENTASI DAN PENGUJIAN	63
A. IMPLEMENTASI SISTEM.....	63

B. PENGUJIAN.....	90
BAB V.....	100
A. HASIL.....	100
B. PEMBAHASAN	108
BAB VI.....	110
A. KESIMPULAN.....	110
B. SARAN	110
DAFTAR PUSTAKA	111

DAFTAR GAMBAR

Gambar 2. 1 Proses Pindah Silang	15
Gambar 2. 2 Proses Mutasi	18
Gambar 2. 3 Siklus Umum Algoritma Genetika.....	19
Gambar 3. 1 Flowchart Alur Penelitian.....	21
Gambar 3. 2 Gambar Alur pembuatan jadwal yang seharusnya.....	22
Gambar 3. 3 Gambar Alur pembuatan jadwal ujian yang berjalan.....	23
Gambar 3. 4 Flowchart sistem berjalan.....	33
Gambar 3. 5 Gambar Flowchart Siklus Pemecahan Masalah dengan Algoritma Genetika	36
Gambar 3. 6 Use Case Diagram.....	47
Gambar 3. 7 Activity Diagram.....	48
Gambar 3. 8 Sequence Diagram Login.....	48
Gambar 3. 9 Sequence Diagram Penjadwalan Ujian dengan Algoritma Genetika.....	49
Gambar 3. 10 Class Diagram	49
Gambar 3. 11 Gambar ERD	53
Gambar 3. 12 Relasi Matakuliah dan Jadwal.....	54
Gambar 3. 13 Relasi Matakuliah dan Jadwal.....	54
Gambar 3. 14 Relasi Waktu Ujian dan Jadwal	54
Gambar 3. 15 Relasi Ruangan dan Jadwal.....	55
Gambar 3. 16 Relasi Jam dan Waktu	55
Gambar 3. 17 Gambar Tampilan Halaman User.....	59
Gambar 3. 18 Gambar Tampilan Halaman Matakuliah.....	60
Gambar 3. 19 Gambar Halaman Pengawas.....	60
Gambar 3. 20 Gambar Tampilan Halaman Waktu.....	61
Gambar 3. 21 Gambar Tampilan Halaman Ruangan	61
Gambar 3. 22 Gambar Tampilan Penjadwal Ujian	62
Gambar 3. 23 Gambar Tampilan Jadwal Ujian.....	62
Gambar 4. 1 gambar implementasi tabel user.....	64
Gambar 4. 2 gambar implementasi tabel matakuliah.....	64
Gambar 4. 3 gambar implementasi tabel waktu.....	65
Gambar 4. 4 gambar implementasi tabel sesi.	65
Gambar 4. 5 gambar implementasi tabel ruang.	65
Gambar 4. 6 gambar implementasi tabel pengawas.....	66
Gambar 4. 7 gambar implementasi tabel jadwal ujian akhir semester.....	66
Gambar 4. 8 gambar login admin.....	66
Gambar 4. 9 Implementasi Antarmuka Beranda.....	67
Gambar 4. 10 Implementasi Antarmuka Data Matakuliah	68
Gambar 4. 11 Implementasi Antarmuka Tambah Matakuliah.....	68
Gambar 4. 12 Implementasi Antarmuka Ubah Matakuliah	69
Gambar 4. 13 Implementasi Antarmuka Data Pengawas.....	69
Gambar 4. 14 Implementasi Antarmuka Tambah Data Pengawas.	70
Gambar 4. 15 Implementasi Antarmuka Ubah Data Pengawas	70
Gambar 4. 16 Implementasi Antarmuka Tampilan Data Ruang.....	71

Gambar 4. 17 Implementasi Antarmuka Tambah Data Ruang	71
Gambar 4. 18 Implementasi Antarmuka Ubah Data Ruang	72
Gambar 4. 19 Implementasi Antarmuka Input Data Jam.....	72
Gambar 4. 20 Implementasi Antarmuka Tambah Data Sesi.....	73
Gambar 4. 21 Implementasi Antarmuka Ubah Data Sesi	73
Gambar 4. 22 Implementasi Antarmuka Tampilan Data Waktu.....	74
Gambar 4. 23 Implementasi Antarmuka Tambah Data Hari	74
Gambar 4. 24 Implementasi Antarmuka Ubah Data Hari.....	75
Gambar 4. 25 Implementasi Antarmuka Input Data Proses Penjadwalan	75
Gambar 4. 26 <i>script</i> program cek login	76
Gambar 4. 27 <i>script</i> Data Matakuliah	77
Gambar 4. 28 <i>script</i> Data Hari	78
Gambar 4. 29 <i>script</i> Data Jam.....	80
Gambar 4. 30 <i>Script</i> Data Ruang	81
Gambar 4. 31 <i>Script</i> Data Pengawas	82
Gambar 4. 32 <i>Script</i> Data Proses Penjadwalan	89
Gambar 4. 33 <i>Script</i> Hasil Jadwal.....	90
Gambar 5. 1 Hasil Jadwal Uji ke satu dengan Aplikasi Algoritma Genetika.....	100
Gambar 5. 2 Hasil penjadwalan dengan jumlah kromosom 10 dan maksimal generasin 25.	101
Gambar 5. 3 Hasil Jadwal Uji ke dua dengan Aplikasi Algoritma Genetika.....	101
Gambar 5. 4 Hasil penjadwalan dengan jumlah kromosom 20 dan maksimal generasin 35.	102
Gambar 5. 5 Hasil Jadwal Uji ke tiga dengan Aplikasi Algoritma Genetika	102
Gambar 5. 6 Hasil penjadwalan dengan jumlah kromosom 30 dan maksimal generasin 35.	103
Gambar 5. 7 Hasil Jadwal Uji ke empat dengan Aplikasi Algoritma Genetika..	103
Gambar 5. 8 Hasil penjadwalan dengan jumlah kromosom 50 dan maksimal generasin 100.	104

DAFTAR TABEL

Tabel 3. 1 Contoh Jadwal UAS Prodi Keperawatan S1	24
Tabel 3. 2 Tabel Jadwal UAS Prodi D3 Keperawatan.....	26
Tabel 3. 3 Tabel Lama Hari Pelaksanaan Ujian.....	26
Tabel 3. 4 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan S1	27
Tabel 3. 5 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan D3	27
Tabel 3. 6 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan D3	28
Tabel 3. 7 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan S1	29
Tabel 3. 8 Tabel Contoh Pengawas yang mengawasi dalam sesi yang sama dan berbeda ruang dalam satu sesi.....	34
Tabel 3. 9 Pengkodean Pegawai	37
Tabel 3. 10 Tabel Pengkodean Jam.....	37
Tabel 3. 11 Tabel Pengkodean Waktu	38
Tabel 3. 12 Tabel Pengkodean Ruang.....	38
Tabel 3. 13 Tabel Pengkodean Makul Keperawatan S1	39
Tabel 3. 14 Tabel Pengkodean Makul Keperawatan D3.....	40
Tabel 3. 15 Tabel Representasi Kromosom Pengawas Hari ke 1	42
Tabel 3. 16 Tabel Proses Evaluasi	44
Tabel 3. 17 Tabel Perhitungan Nilai Fitness pada child (offspring).....	45
Tabel 3. 18 Tabel Kumpulan Individu & offspring pada hari 1.....	45
Tabel 3. 19 Tabel Individu Terpilih pada Generasi Berikutnya.....	46
Tabel 3. 20 Tabel Entitas	50
Tabel 3. 21 Tabel Penentuan Entitas dan Atribut	50
Tabel 3. 22 Tabel Rancangan Tabel Matakuliah	55
Tabel 3. 23 Tabel Rancangan Tabel Pengawas.....	56
Tabel 3. 24 Tabel Rancangan Tabel Ruang	56
Tabel 3. 25 Tabel Waktu Ujian	57
Tabel 3. 26 Tabel Rancangan Tabel Jadwal.....	57
Tabel 5. 1 Tabel Penentuan Skor	106
Tabel 5. 2 Tabel Presentase Kriteria Skor.....	107
Tabel 5. 3 Tabel Hasil Kueisioner	107

BAB 1

PENDAHULUAN

A. Latar Belakang

Pembuatan jadwal ujian akhir semester adalah kegiatan rutin yang dilakukan oleh perguruan tinggi pada tiap semester. Di perguruan tinggi, program penjadwalan ujian merupakan salah satu hal penting dalam proses belajar mengajar, karena semua kegiatan ujian semester bergantung pada jadwal yang ada, sehingga harus disusun dengan benar dan diperbaiki setiap akan diadakan ujian pada tiap semester, sehingga nantinya tidak terjadi benturan jadwal ujian semester. Pembuatan jadwal Ujian membutuhkan banyak waktu untuk menghasilkan jadwal yang baik dan tepat sasaran.

Sistem penjadwalan pada sebuah Universitas terutama penjadwalan ujian semester, merupakan hal yang kritis dan perlu diperhitungkan dengan baik karena harus mempertimbangkan berbagai keterbatasan dan syarat. Jadwal ujian semester di universitas harus dibuat tiap semester, dan jadwal tersebut tidak akan dapat digunakan kembali pada semester yang sama ditahun berikutnya. Karena itulah, permasalahan penjadwalan mata kuliah merupakan permasalahan yang kompleks.

Pembuatan jadwal Ujian di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang dirancang oleh ketua program studi. Penjadwalan ujian semester di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang terdapat pada terbatasnya ruang ujian dan pengawas ujian, akan tetapi jumlah kelas perkuliahan sangat banyak. Pembuatan jadwal ujian selama ini masih menggunakan cara manual dengan menggunakan perangkat lunak Microsoft Excel yang rentan akan kesalahan. Penjadwalan yang manual menggunakan Microsoft Exel akan memakan waktu yang lama karena banyaknya factor yang harus dipertimbangkan.

Faktor yang menjadi pertimbangan dalam merancang jadwal ujian semester antara lain adalah jumlah mata kuliah, jumlah ruangan kelas, jumlah pengawas ujian dan waktu ujian. Permasalahan penjadwalan ujian dua prodi

yang memakan waktu lama karena masing-masing prodi membuat jadwal masing-masing sehingga salah satu prodi menunggu jadwal lainnya. Distribusi jadwal ujian diharapkan dapat merata setiap harinya untuk setiap kelas dan merata juga bagi pengawas ujian agar tidak menumpuk jadwal ujian dalam satu hari. Selain itu sebuah jadwal dirancang sedemikian rupa agar pengawas ujian tidak mengawasi ujian yang berbeda pada waktu yang bersamaan. Sebuah penjadwalan dikatakan baik apabila dapat memberikan solusi terhadap factor-faktor tersebut, tentu saja tidak semua sistem penjadwalan memiliki permasalahan yang sama, karena disesuaikan dengan kebutuhan instansi ataupun tempat yang akan dilakukan penjadwalan.

Permasalahan penjadwalan diatas sering terjadi di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Permasalahan-permasalahan yang terjadi diharapkan mampu memecahkan setiap detail masalah yang muncul dalam proses penjadwalan di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Banyak metode yang dapat digunakan dalam menyelesaikan masalah penjadwalan seperti metode *graph coloring*, metode *fuzzy logic*, metode algoritma genetika, metode *auto generate time table*, metode *tabu search*, dan *Partical swarm optimization*. Penelitian ini menggunakan metode Algoritma Genetika untuk memecahkan masalah penjadwalan, karena dengan metode algoritma genetika merupakan salah satu metode optimasi yang kuat dan bisa digunakan pada berbagaimacam studi kasus, baik kasus yang sederhana hingga kasus yang rumit karena menggunakan teori evolusi (Pradyana, Sunaryono, dan munif 2012)

Selain itu dalam Algoritma genetika semua modul berjalan dengan baik dan jadwal yang dihasilkan lebih presisi dengan pengalokasian jam ujian, ruang kelas dan pengawas ujian yang tepat. Berdasarkan hasil ini maka dapat disimpulkan bahwa system yang dibangun telah berjalan dengan baik sesuai dengan kebutuhan yang telah diidentifikasi diawal (Afrizal Nehemia Toscan, Rusdianto Roestam 2017). Penggunaan metode algoritma genetika pada penelitian ini diharapkan dapat membantu mengoptimalakan

dan memenuhi segala permasalahan dalam ujian semester. Selain itu dapat membuat system melakukan proses penjadwalan secara otomatis.

Berdasarkan latar belakang diatas, maka penelitian ini berusaha merancang sebuah system penjadwalan menggunakan metode algoritma genetika dan mengangkatnya menjadi sebuah penelitian yang berjudul Otomatisasi Sistem Penjadwalan Ujian Semester Menggunakan Metode Algoritma Genetika (Studi Kasus : Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang).

B. Perumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana jadwal ujian dua prodi dibuat secara bersamaan menggunakan penjadwalan ujian otomatis?
2. Bagaimana penjadwalan ujian otomatis dapat menjamin seorang pengawas tidak mengawas ujian dalam waktu yang bersamaan?
3. Bagaimana penjadwalan ujian otomatis menjamin 1 matakuliah hanya ujian satukali dalam 1 hari?
4. Bagaimana algoritma genetika membantu menyelesaikan permasalahan yang ada dalam penjadwalan ujian otomatis?

C. Batasan Masalah

Dalam suatu penelitian dapat muncul permasalahan yang meluas. Agar permasalahan dalam sebuah penelitian tidak meluas maka dibutuhkan suatu batasan masalah antara lain :

1. Sampel data yang digunakan adalah Program Studi Keperawatan S1 dan Program Studi Keperawatan D3 Semester Genap tahun akademik 2018/2019.
2. Diasumsikan waktu ujian per-matakuliah adalah 75 sampai 90 menit.
3. Diasumsikan pengawas bersedia mengawasi di ruang manapun yang tersedia dan pada waktu yang telah ditentukan.

4. Sistem penjadwalan ini hanya untuk Program Studi dengan kelas regular
5. Metode yang digunakan adalah Algoritma Genetika.

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membuat penjadwalan ujian otomatis dua prodi secara bersamaan dengan menjamin seorang pengawas tidak mengawasi ujian dalam waktu bersamaan.
2. Membuat sistem penjadwalan ujian otomatis dengan menerapkan metode algoritma genetika.

E. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat membantu proses penjadwalan yang dilakukan di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang mempermudah pembuatan jadwal ujian.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Relevan

1. Penelitian yang dilakukan oleh Afrizal Nehemia Toscany, Rusdianto Roestam (2017) yang berjudul Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetika (Studi Kasus : Pascasarjana Universitas Jambi) menyatakan bahwa Saat ini penjadwalan kuliah di Pascasarjana Universitas Jambi masih terdapat beberapa kendala seperti pembuatan jadwal yang relatif lama, jadwal yang bentrok dan jadwal kuliah yang tidak sesuai dengan waktu ketersediaan dosen. Oleh karena itu penulis memberikan solusi berupa sistem penjadwalan kuliah menggunakan algoritma genetik dengan bahasa pemrograman PHP dan DBMS MySQL. Untuk metode pengembangan sistem yang digunakan adalah waterfall. Dimana hasil jadwal kuliah yang dihasilkan dengan sistem yang telah dibangun ini menjadi lebih presisi dengan pengalokasian jam mengajar, ruang kelas dan dosen yang tepat. Dari hasil pengujian juga dapat disimpulkan bahwa implementasi algoritma genetik sudah sesuai dengan kebutuhan untuk mendukung proses penjadwalan, sehingga penyusunan jadwal bisa dilakukan dengan lebih cepat.
2. Penelitian yang dilakukan oleh Lian Aga Aditya, Windha Mega PD (2017) yang berjudul Implementasi Algoritma Genetika Untuk Penjadwalan Mata Pelajaran Pada LMS Getsmart menyatakan bahwa Penjadwalan dalam kegiatan belajar mengajar di institusi pendidikan merupakan hal yang rumit jika dilakukan secara manual. Karena untuk mengatur jadwal melibatkan banyak komponen seperti jumlah guru, kelas, dan ketersediaan waktu sesuai jadwal yang bisa digunakan. Dengan menganalisa permasalahan ini, sebuah sistem yang dapat membantu kita untuk mengoptimalkan dan secara otomatis menghasilkan jadwal kerja sangat dibutuhkan. Pendekatan untuk mengoptimalkan proses penentuan waktu adalah dengan menggunakan Algoritma Genetika. Algoritma ini merupakan pendekatan komputasi dari prinsip seleksi alam dari teori evolusi Charles Darwin dan teori pewarisan Mendel. Studi kasus penelitian

ini dilakukan di LMS GETSMART yang merupakan platform e-learning yang belum memiliki fitur penetapan waktu. Melalui metode algoritma genetika dan dengan menggunakan data kendala yang ada, sistem ini mampu menghasilkan jadwal yang paling optimal berdasarkan nilai *fitnessnya*, karena algoritma ini menggunakan kombinasi prinsip seleksi alam dan prinsip ketegaran untuk mendapatkan hasil seperti yang diharapkan.

3. Penelitian yang dilakukan oleh Ivan, Stephanus Raphael, Halim Agung (2018) yang berjudul Aplikasi Penjadwalan mata kuliah di SMAN 31 Menggunakan Algoritma Genetika Berbasis Web menyatakan bahwa Sekolah Menengah Atas Negeri 31 memiliki kelas sebanyak 29 kelas yang masing-masing memiliki 3 jurusan yaitu IPA, IPS dan Bahasa. Masing masing jurusan memiliki jumlah kelas yang cukup banyak. Sehingga pembuatan jadwal mata pelajaran oleh pihak sekolah memerlukan waktu yang cukup lama dan mengganggu kelancaran proses belajar dan mengajar. aplikasi penjadwalan mata pelajaran menggunakan algoritma genetika merupakan solusi yang tepat untuk SMAN31. Algoritma Genetika adalah Proses seleksi alamiah yang melibatkan perubahan gen yang terjadi pada individu melalui proses perkembang-biakan yang menjadi proses dasar dan menjadi perhatian utama. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah Metode Waterfall. Hasil dari penelitian ini adalah aplikasi penjadwalan SMAN 31 Jakarta yang dapat menyusun jadwal yang baik secara otomatis. Kesimpulan dari penelitian ini adalah algoritma genetika dapat diimplementasikan pada aplikasi penjadwalan dibuktikan dengan menguji 30 contoh jadwal yang tersusun oleh aplikasi tanpa ada yang bertabrakan satu sama lain.

Dari ketiga penelitian diatas dapat disimpulkan bahwa penyelesaian masalah penjadwalan yang masih manual dan begitu rumit dapat diselesaikan dengan Algoritma genetika. Masing-masing penerapan dari aplikasi ini menggunakan metode yang sama yaitu metode Algoritma Genetika. Penulis ingin membuat suatu sistem yang dapat mengatur penyusunan jadwal yang baik sesuai dengan sumber daya dan batasan

yang ada dengan menggunakan Algoritma Genetika. Metode ini digunakan pencarian *heuristic* yang didasarkan atas mekanisme evolusi biologis. Setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma ini adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom. Metode yang dapat disebut algoritma genetika adalah metode yang setidaknya memiliki komponen seperti populasi dari kromosom, seleksi berdasarkan *fitness*, *crossover* untuk memproduksi *offspring* baru, dan random mutasi pada *offspring*. Dari cara tersebut yang nantinya dapat menghasilkan penyusunan jadwal yang cukup baik dan akurat yang bermanfaat buat Kaprodi.

B. Penjelasan Teoritis Masing – Masing Variabel Penelitian

1. Penjadwalan

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan di mana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas. Menurut (Kenneth R. Baker, 1974) penjadwalan adalah proses untuk melakukan tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan.

Pada dasarnya penjadwalan mencakup pengurutan aktivitas, pengalokasian aktivitas pada fasilitas dan pemetaan aktivitas menurut urutan waktu. Menurut (Farida, 2008), Penjadwalan merupakan kumpulan kebijaksanaan dan mekanisme dalam system operasi yang berhubungan dengan urutan kerja yang dilakukan system computer.

Terdapat batasan atau persyaratan (*constraints*) dalam penyusunan penjadwalan ujian akhir semester. *Constraints* sendiri merukan suatu syarat tidak boleh terjadi pelanggaran terhadap kendala yang ditetapkan agar dapat menghasilkan susunan penjadwalan yang baik. Beberapa constraint tersebut , yaitu:

Pengawas tidak boleh dijadwalkan lebih dari satu kali pada waktu yang bersamaan.

Satu kelas dan ruang tidak boleh dijadwalkan lebih dari satu kali pada waktu yang bersamaan.

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

2. Algoritma Genetika

Algoritma Genetika pertama kali dikembangkan tahun 1970-an oleh Jhon Holland. Tujuan yang ingin dicapai Holland saat itu adalah mengabstraksikan poses-proses evolusi yang terjadi di alam dan mendesain suatu software yang prinsip kerjanya meniru proses-proses evolusi. Hasilnya, algoritma genetika ternyata dapat menyelesaikan masalah-masalah yang tidak dapat diselesaikan dengan menggunakan perhitungan matematika biasa (Arkeman Yandra, Kudang Boro Seminar, 2012).

Algoritma genetika adalah suatu teknik pencarian dan teknik optimasi yang cara kerjanya meniru proses evolusi dan perubahan struktur genetic pada makhluk hidup (Arkeman Yandra, Kudang Boro Seminar, 2012). Algoritma ini sering digunakan untuk mencari solusi optimal baik kasus yang sederhana sampai kasus yang rumit. Algoritma genetik bekerja pada suatu populasi yang dibentuk oleh sebuah kasus yang direpresentasikan sebagai kromosom dan akan dievaluasi untuk memperoleh seberapa nilai optimalnya. Apabila pada generasi tersebut belum ditemukan kasus yang memiliki nilai paling optimal, maka akan dibentuk populasi baru dengan cara melakukan seleksi kepada individu yang nilai optimalnya kecil dan digantikan dengan individu baru melalui proses reproduksi .

Hal-hal yang dilakukan dalam menggunakan algoritma genetika adalah (Sitanggang, 2015):

Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.

Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu atau baik tidaknya solusi yang didapatkan.

Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.

Menentukan proses seleksi yang akan digunakan.

Menentukan proses perkawinan silang (*cross-over*).

Mutasi gen yang akan digunakan.

3. Struktur Umum Algoritma Genetika

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti terjadi pada makhluk hidup. Inisialisasi populasi awal dilakukan untuk menghasilkan solusi awal dari suatu permasalahan algoritma genetika. Inisialisasi ini dilakukan secara acak sebanyak jumlah kromosom yang diinginkan. Selanjutnya dihitung nilai *fitness* dan seterusnya dilakukan seleksi dengan menggunakan metode roda *roulette*, tournament atau ranking.

Kemudian dilakukan perkawinan silang (*crossover*) dan mutasi. Setelah melalui beberapa generasi maka algoritma ini akan berhenti sebanyak generasi yang diinginkan. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator reproduksi, operator *crossover* (persilangan) dan operator mutasi. Ada dua hal penting yang harus dilakukan pada awal proses Algoritma Genetika. Pertama, pendefinisian atau pengkodean kromosom yang merupakan solusi yang masih berbentuk simbol. Kedua, penentuan fungsi *fitness* atau fungsi obyektif. Dua hal ini berperan penting dalam algoritma genetika untuk menyelesaikan suatu masalah (Maharani, 2013).

4. Komponen – Komponen Algoritma Genetika

Ada beberapa komponen dalam algoritma genetika, yaitu :

a. Pengkodean

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dikodekan terlebih dahulu kedalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengodekan informasi yang disimpan didalam individu atau kromosom.

Berdasarkan jenis simbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut (Maharani, 2013):

- 1) Pengkodean biner merupakan cara pengkodean yang paling umum digunakan, karena pengkodean ini merupakan yang pertama kali digunakan dalam algoritma genetika oleh Holland (Cordon et al, 2001). Pengkodean biner dinyatakan dalam kromosom biner.
- 2) Pengkodean bilangan bulat. Pengkodean ini baik digunakan untuk masalah optimasi kombinatorial. Dengan pengkodean bilangan bulat, ukuran kromosom menjadi lebih sederhana dan tidak terlalu panjang.
- 3) Pengkodean bilangan riil. (Gen dan Cheng 2000) menyatakan bahwa pengkodean bilangan riil baik digunakan untuk masalah optimasi fungsi dan optimasi kendala.
- 4) Pengkodean struktur data adalah model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah yang lebih kompleks seperti perencanaan trajektori robot dan masalah pewarnaan grap.

b. Inisialisasi Populasi Awal

Inisialisasi populasi awal merupakan suatu metode untuk menghasilkan kromosom-kromosom awal. Jumlah individu pada populasi awal merupakan masukan dari pengguna. Setelah jumlah individu pada populasi awal ditentukan, dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi dilakukan secara acak, namun tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

c. Fungsi Evaluasi (Fungsi *Fitness*)

Fungsi evaluasi dalam algoritma genetika merupakan sebuah fungsi yang memberikan penilaian kepada kromosom (*fitness value*) untuk dijadikan suatu acuan dalam mencapai nilai optimal pada algoritma genetika. Nilai *fitness* ini kemudian menjadi nilai bobot suatu kromosom. Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum fungsi *fitness* ditentukan dari fungsi objektif dengan nilai yang tidak negatif, jika ternyata nilai dari fungsi objektif bernilai negatif maka perlu ditambahkan suatu konstanta x agar nilai *fitness* yang terbentuk tidak bernilai negatif (Putra, 2009).

Didalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, jika solusi yang akan dicari adalah memaksimalkan fungsi h (dikenal sebagai masalah maksimasi) sehingga nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f = h$ (dimana f adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi h (masalah minimasi), maka fungsi h tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang bisa digunakan adalah $f = 1/h$, yang artinya semakin kecil nilai h , semakin besar nilai f . Tetapi hal ini akan menjadi masalah jika

h bisa bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasinya, h perlu ditambah sebuah bilangan yang dianggap kecil [0-1] sehingga nilai *fitness*nya menjadi :

$$f = \frac{1}{h} + a$$

dengan a adalah bilangan yang kecil dan bervariasi [0-1] sesuai dengan masalah yang akan diselesaikan (Suyanto,2005). Oleh karena itu fungsi *fitness* menjadi masalah atau penentu utama keberhasilan algoritma genetika. Didalam penelitian ini batasan atau *constraint* dalam penyusunan jadwal proyek yang dijadikan fungsi objektifnya yaitu meminimumkan pelanggaran terhadap *constraint* yang telah ditentukan.

d. Seleksi

Seleksi merupakan proses pemilihan orang tua untuk reproduksi (biasanya didasarkan pada nilai *fitness*). Seleksi bertujuan untuk memberikan kesempatan reproduksi yang paling besar bagi anggota populasi yang paling baik. Ada beberapa metode yang bisa digunakan dalam tahap seleksi, diantaranya :

1) *Range Base Fitness*

Populasi diurutkan berdasarkan nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan dan tidak dipengaruhi oleh nilai objektifnya.

2) *Roulette Wheel Selection* (Seleksi Roda Roulette)

Individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki urutan yang sama dengan ukuran *fitness*. Sebuah bilangan acak dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Cara penyelesaian metode ini meniru permainan roda roulette, dimana

prosedur seleksi dimulai dengan memutar roda roulette sebanyak n , setiap waktu dipilih satu kromosom sebagai induk untuk menghasilkan kromosom baru. Metode roulette-wheel selection sangat mudah diimplementasikan dalam pemrograman. Pertama, dibuat interval nilai kumulatif (dalam interval $[0,1]$) dari nilai *fitness* masing-masing kromosom dibagi total nilai *fitness* dari semua kromosom. Sebuah kromosom akan terpilih jika bilangan random yang dibangkitkan berada dalam interval akumulatifnya. Metode ini merupakan salah satu yang paling umum digunakan (Goldsberg, 1989 dikutip dari Riza Arifudin, 2011).

3) *Stochastic Universal Sampling*

Individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga hingga tiap-tiap segmen individu memiliki urutan yang sama dengan ukuran *fitness*. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut.

4) *Local Selection* (Seleksi Lokal)

Tiap individu yang berada pada konstrain tertentu disebut nama lingkungan lokal. Interaksi antar individu hanya dilakukan di dalam wilayah tersebut. Lingkaran tersebut ditetapkan sebagai struktur dimana populasi tersebut terdistribusi.

5) *Truncation Selection* (Seleksi dengan pemotongan)

Individu diurutkan berdasarkan *fitness*, hanya individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang (trunk) yang mengidentifikasi ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50%-100%.

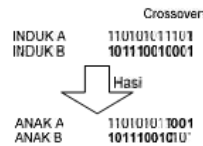
6) *Tournament Selection* (Seleksi dengan turnamen)

Dalam bentuk paling sederhana, metode ini mengambil dua kromosom secara random dan kemudian menyeleksi salah satu yang bernilai *fitness* paling tinggi untuk menjadi orang tua pertama. Cara yang sama dilakukan lagi untuk mendapatkan orang tua yang kedua. Metode tournament selection yang lebih rumit adalah dengan mengambil m kromosom secara random. Kemudian kromosom bernilai *fitness* tertinggi dipilih sebagai orang tua pertama jika bilangan random yang dibangkitkan kurang dari suatu nilai batas yang ditentukan p dalam interval $[0,1]$. Pemilihan orang tua akan dilakukan secara random dari $m - 1$ kromosom yang ada jika bilangan random yang dibangkitkan lebih dari atau sama dengan p . Pada tournament selection, variabel m adalah tournament size dan p adalah tournament probability. Biasanya m diset sebagai suatu nilai yang sangat kecil, misal 4 atau 5. Sedangkan p biasanya diset sekitar 0,75. (Suyanto, 2005)

e. Pindah Silang (*Crossover*)

Pindah silang atau *crossover* adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom. *Crossover* menghasilkan kromosom baru yang disebut kromosom anak (*offspring*). *Crossover* bertujuan untuk menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya (Arifudin, 2011).

Crossover merupakan operator genetik utama, yang beroperasi pada dua kromosom dalam suatu waktu dan menghasilkan *offspring* dengan mengkombinasikan kedua fitur-fitur kromosom (Fadlisyah dkk, 2009). Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom. Contoh *crossover* dapat dilihat pada gambar berikut:



Gambar 2. 1 Proses Pindah Silang

Proses *crossover* dilakukan dengan cara memilih dua induk dengan kualitas yang baik, setelah itu dilakukan proses ekstraksi kromosom setiap induk. Titik potong ditentukan secara acak, kemudian dilakukan pertukaran bit-bit kromosom disebelah kanan titik kromosom sehingga terbentuk keturunan yaitu anak A dan B. Kromosom anak sebagian besar masih mewarisi kromosom induk tetapi sebagian lagi sudah terjadi pertukaran materi genetik antar kromosom.

Pindah silang bisa juga berakibat buruk jika ukuran populasinya sangat kecil. Dalam suatu populasi yang sangat kecil, suatu kromosom dengan gen-gen yang mengarah ke solusi akan sangat cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah ini digunakan suatu aturan bahwa pindah silang hanya bisa dilakukan dengan suatu probabilitas tertentu (probabilitas *crossover*). Artinya pindah silang bisa dilakukan hanya jika suatu bilangan random $[0,1]$ yang dibangkitkan kurang dari probabilitas *crossover* (P_c) yang ditentukan. Pada umumnya P_c diset mendekati 1, misalnya 0.8 (Suyanto,2005). Probabilitas *crossover* (P_c) bertujuan untuk mengendalikan operator *crossover*. Jika n adalah banyaknya string pada populasi, maka sebanyak $(P_c) \times n$ string akan mengalami *crossover*. Semakin besar nilai (P_c), semakin cepat pula string baru muncul dalam populasi. Dan juga jika (P_c) terlalu besar, string yang merupakan kandidat solusi terbaik mungkin dapat hilang lebih cepat pada generasi berikutnya.

Proses *crossover* memiliki nilai kemungkinan yang besar dalam satu siklus algoritma genetika karena tujuan utamanya adalah membentuk keragaman individu, semakin tinggi probabilitas

crossover maka semakin cepat keragaman terbentuk. *Crossover* dapat dilakukan dengan beberapa cara yang berbeda, diantaranya:

1) Penyilangan Satu Titik (*One-point Crossover*)

Metode *crossover* ini yang sering digunakan pada algoritma genetika. Pada penyilangan satu titik, posisi penyilangan k ($k=1,2,3,\dots,n$) dengan n = panjang kromosom yang diseleksi secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

2) Penyilangan Banyak Titik (*Multi-point Crossover*)

Untuk kromosom yang sangat panjang, misalkan 1000 gen, mungkin saja diperlukan beberapa titik potong. Penyilangan dilakukan sebanyak m ($m=1, 2, 3,\dots,n$) dengan posisi penyilangan k ($k=1,2,3,\dots,n$) yang ditentukan secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

3) Penyilangan Seragam (*Uniform Crossover*)

Dibentuk suatu kromosom sepanjang kromosom induk dengan bit-bit yang dipilih secara acak, kemudian penyilangan dilakukan sebanyak m ($m=1, 2, 3,\dots,n$) dengan posisi penyilangan k ($k=1,2,3,\dots,n$) yang ditentukan secara acak. Pada titik tersebut dilakukan pertukaran antar kromosom induk untuk menghasilkan anak.

f. Mutasi

Mutasi merupakan proses mengubah secara acak nilai dari satu atau beberapa gen dalam suatu kromosom (Haupt, 2004 dikutip dari Riza Arifudin, 2011). Mutasi adalah operator algoritma genetika

yang bertujuan untuk membentuk individu-individu yang baik atau memiliki kualitas diatas rata-rata.

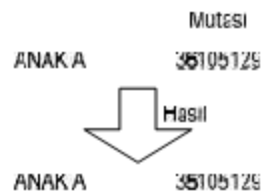
Selain itu mutasi dipergunakan untuk mengembalikan kerusakan materi genetic akibat proses *crossover*. Di dalam algoritma genetika, mutasi melakukan peran vital berikut (Fadlisyah dkk, 2009) :

- 1) Menempatkan kembali gen-gen yang hilang dari populasi sepanjang proses seleksi, agar mereka dapat dilibatkan kembali pada konteks yang berikutnya.
- 2) Menyediakan gen-gen yang tidak hadir pada populasi awal.

Pada mutasi terdapat satu parameter yang sangat penting, yaitu probabilitas mutasi (P_m) yang bertujuan untuk mengendalikan operator mutasi. Probabilitas mutasi didefinisikan sebagai persentasi dari jumlah total gen dalam populasi yang akan mengalami mutasi. Disetiap generasi diperkirakan terjadi mutasi sebanyak $(P_m) \times n$ sring. Pada seleksi alam murni, mutasi jarang sekali muncul sehingga probabilitas mutasi yang digunakan umumnya kecil, lebih kecil dari probabilitas *crossover*. P_m biasanya diset antara $[0-1]$, misalnya 0.1 (Suyanto,2005). Misalkan offspring yang terbentuk adalah 100 dengan jumlah gen setiap kromosom adalah 4 dan peluang mutasi adalah 0.10, maka diharapkan terdapat 40 kromosom dari 400 gen yang ada pada populasi tersebut akan mengalami mutasi.

Mutasi yang digunakan dalam penelitian ini adalah mutasi dengan pengkodean bulat. Mutasi gen ini dilakukan dengan cara pemilihan nilai secara acak. Suatu gen yang terpilih untuk dimutasi nilainya diganti dengan nilai baru yang dibangkitkan secara acak dalam interval nilai-nilai gen yang diizinkan. Misalnya, jika nilai-nilai gen berada dalam interval $[0.9]$, maka gen baru yang dibangkitkan secara acak juga berada dalam interval $[0.9]$. Nilai

gen baru yang dihasilkan bisa saja dibatasi dengan aturan berbeda dengan nilai lama.



Gambar 2. 2 Proses Mutasi

g. *Elitism*

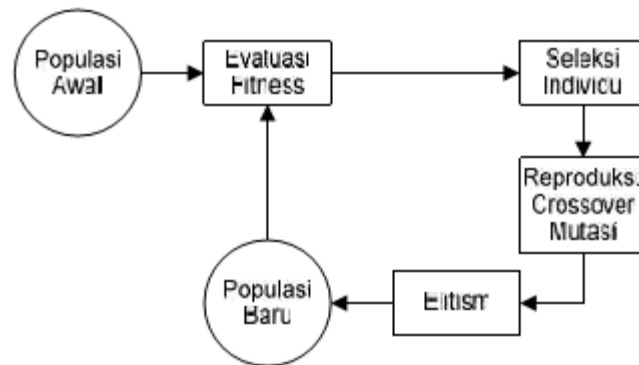
Karena seleksi dilakukan secara acak (random), maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu yang memiliki nilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness* menurun) karena proses *crossover*. Untuk menjaga agar individu yang bernilai *fitness* tertinggi tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa buah duplikatnya. Proses ini dikenal sebagai elitism.

5. Siklus Umum Algoritma Genetika

Langkah Umum pada algoritma genetika adalah sebagai berikut (Putra, 2009):

- a. Melakukan inisialisasi populasi kromosom dengan solusi secara acak (random).
- b. Melakukan evaluasi setiap kromosom dalam populasi menggunakan persamaan fungsi evaluasi (*fitness function*).
- c. Memilih sebagian anggota populasi sebagai solusi yang sesuai dengan induknya untuk generasi selanjutnya.
- d. Menciptakan solusi (keturunan) baru dengan mengawinkan solusi dari induknya dengan cara *crossover* dan mutasi.

- e. Membuang atau menghapus anggota populasi lama yang tidak produktif untuk membuat ruang solusi yang baru agar dapat masuk kedalam populasi.
- f. Jika aturan pemberhentian terpenuhi, berhenti dan keluarkan kromosom yang paling baik. Jika tidak kembali ke langkah 3.



Gambar 2. 3 Siklus Umum Algoritma Genetika.

6. PHP

Secara khusus PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, bisa menampilkan database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun perlu diketahui bahwa PHP sebenarnya bisa dipakai secara command line. Artinya skrip PHP dapat dijalankan tanpa melibatkan web server maupun browser. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi tool yang disebut “Personal Home Page”. Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa

berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

7. MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu, ia bersifat Open Source pada berbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersil). MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

C. LANDASAN TEORI

Berdasarkan ketiga penelitian yang relevan diatas dan penjabaran variable-variabel yang berkaitan dengan penelitian ini, dapat diambil kesimpulan bahwa semua penelitian relevan terkait membahas tentang sistem penjadwalan secara otomatis yang dilakukan penulis, pada penelitian ini, penulis membuat sistem penjadwalan ujian semester otomatis yang bisa memunculkan jadwal ujian dan hasil akhir penjadwalan bisa dilihat lewat halaman web. Penulis membuat sistem ini bertujuan agar pihak Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang dapat membuat jadwal secara otomatis dengan menggunakan metode yaitu Algoritma Genetika.

Untuk penentuan Penjadwalan yang secara otomatis ini, penulis memilih metode Algoritma Genetika dalam menentukan jadwal ujian semester. Dengan adanya sistem ini yang diharapkan akan mempermudah fakultas, Kaprodi, Universitas Muhammadiyah Magelang khususnya fakultas Ilmu Kesehatan.

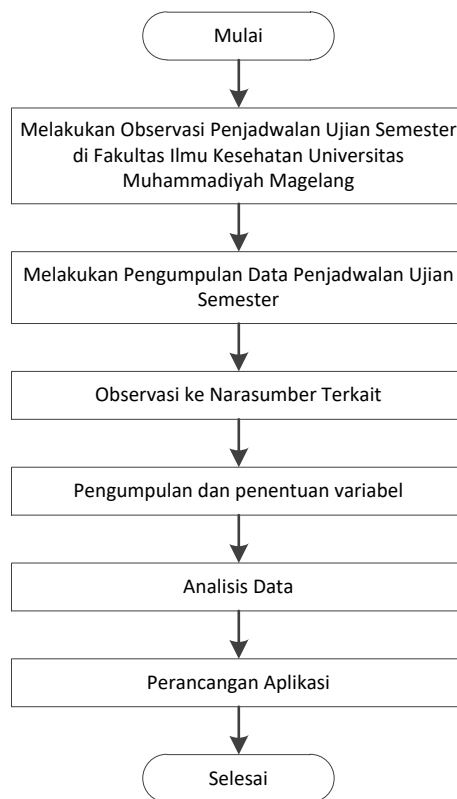
BAB III

ANALISIS DAN PERANCANGAN SISTEM

A. Metode Penelitian

1. Alur Penelitian

Penelitian dilakukan dengan merancang dan membangun sebuah sistem ujian semester dan mengimplementasikan teknik optimasi untuk penjadwalan ujian semester dengan Algoritma Genetika



Gambar 3. 1 Flowchart Alur Penelitian

2. Observasi

Pengumpulan data dengan melakukan pengamatan langsung yang diawali dengan observasi deskriptif yaitu mengamati secara Umum Obyek penelitian terkait dengan system yang telah dijalankan, kemudian dengan observasi terfokus yaitu pengamatan dengan memfokuskan pada pokok – pokok permasalahan yang diteliti mengenai kendala yang dihadapi pada system, dan terakhir dengan observasi menggabungkan kembali pokok –

pokok permasalahan terhadap obyek yang diteliti sehingga memperoleh gambaran yang jelas.

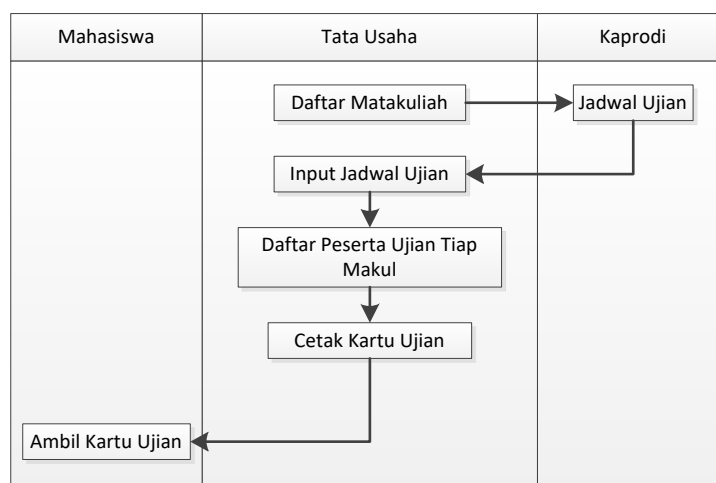
3. Wawancara

Pengumpulan data dengan melakukan Tanya jawab kepada Kaprodi yang memahami dan bertanggung jawab terhadap proses pembuatan ujian semester. Narasumber dalam penelitian ini adalah Kaprodi , sekretaris prodi, dan staf Tata Usaha. Hasil yang diperoleh dari wawancara meliputi proses awal pembuatan jadwal sampai selesai.

4. Dokumentasi

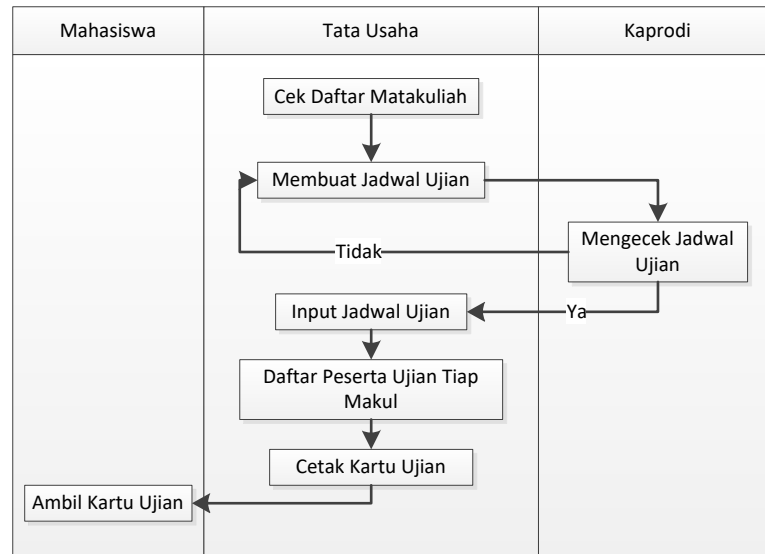
Pengumpulan data dengan mendokumentasikan jadwal, serta kelengkapan lain yang terlibat dengan penjadwalan ujian semester.

- Pembuatan jadwal dua prodi memakan waktu 2 hari.
- Permasalahan penjadwalan dua prodi yang memakan waktu lama karena masing-masing prodi membuat jadwal masing-masing sehingga salah satu prodi menunggu jadwal yang lainnya.
- Alokasi waktu ujian kurang lebih 75 menit sampai 90 menit.
- Alur pembuatan jadwal UAS di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang yang seharusnya.



Gambar 3. 2 Gambar Alur pembuatan jadwal yang seharusnya

- Alur pembuatan jadwal UAS di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang yang seharusnya.



Gambar 3. 3 Gambar Alur pembuatan jadwal ujian yang berjalan.

f. Proses Penjadwalan

1. Matakuliah yang diujikan

Dalam ujian akhir semester ini ada berapa jumlah mata kuliah setiap tingkatan (angkatan)

- Tingkat 1 = a, b, c, d
- Tingkat 2 = e, f, g, h
- Tingkat 3 = i, j, k, l

2. Sesi

Penentuan jadwal ujian dalam satu hari bisa dibuat beberapa sesi, ini dilihat dari jumlah mata kuliah yang ada. Di Prodi Keperawatan S1 dan D3 dibuat satu hari terdapat tiga sesi waktu ujian.

3. Hari

Hari penyelenggaraan ujian dibuat seefektif dan seefisien mungkin, dengan batasan range maksimalnya 2 minggu. Waktu 2 minggu ini adalah hari efektif masa ujian sesuai yang ditentukan berdasarkan kalender akademik yaitu hari Senin sampai dengan hari Sabtu.

4. Waktu

Penyelenggaraan ujian setiap sesi mata kuliah dapat ditentukan sesuai kebijakan masing-masing prodi. Apabila Kaprodi

memberikan kebijakan waktu ujian 90 menit per mata kuliah, dengan 3 sesi waktu ujian ditambah waktu istirahat 15 menit, maka bisa dihitung waktu ujian sebagai berikut:

- Sesi 1 : 08.00 - 09.30
- Sesi 2 : 09.45 - 11.15
- Sesi 3 : 13:00 – 14.30

5. Aturan penjadwalan terdiri dari:

- a. Pengawas tidak boleh mengawasi dalam sesi yang sama dan berbeda ruang.
- b. ruangan ujian tidak boleh digunakan untuk ujian dalam sesi yang sama dan berbeda mata kuliah.

6. Ruang Ujian

Ruangan yang digunakan untuk tempat ujian adalah ruang 305 sampai 312. Satu ruangan memiliki kapasitas untuk 30 orang peserta ujian.

7. Pengawas Ujian

Pengawas ujian merupakan staff yang bekerja di Fakultas Ilmu Kesehatan. Terdapat 5 staff dan 1 pembantu pelaksana di Fakultas Ilmu Kesehatan UMMagelang.

g. Contoh Jadwal UAS Prodi Keperawatan S1

Tabel 3. 1 Contoh Jadwal UAS Prodi Keperawatan S1

JADWAL ASSESSMENT AKHIR SEMESTER KEPERAWATAN S1							
SEMESTER GENAP TAHUN AJARAN 2018/2019							
NO	HARI	TANGGAL	WAKTU	TK	RUANG	MATA KULIAH	PAWAS
1	Senin	1-Jul-19	08.00 - 09.30	I	308,312	PENDIDIKAN DAN PROMOSI KESEHATAN	BD
			10.00 - 11.30	I	308,312	ILMU DASAR KEPERAWATAN II	BD
2	Selasa	2-Jul-19	08.00 - 09.30	I	308,312	PSIKOLOGI & BUDAYA DLM KEPERAWATAN	AB

			10.00 - 11.30	I	308,312	KOMUNIKASI DALAM KEPERAWATAN I	BE
3	Rabu	3-Jul-19	08.00 - 09.30	I	308,312	BAHASA INGGRIS KEPERAWATAN II	BD
			10.00 - 11.30	I	308,312	KEPERAWATAN DASAR II	BA
4	Kamis	4-Jul-19	08.00 - 09.30	I	308,312	KEWARGANEGARAAN	AB
			10.00 - 11.30	I	308,312	KONSEP DASAR KEPERAWATAN II	BE
5	Senin	8-Jul-19	08.00 - 09.30	II	308,312	KEPERAWATAN MEDIKAL BEDAH II	BC
			10.00 - 11.30	IV	308,312	TERAPI KOMPLEMENTER	DE
6	Selasa	9-Jul-19	08.00 - 09.30	II	308,312	KEPERAWATAN GAWAT DARURAT	BE
			10.00 - 11.30	II	308,312	NURSING IN ISLAMIC	AD
7	Rabu	10-Jul-19	08.00 - 09.30	II	308,312	BAHASA INGGRIS KEPERAWATAN III	CE
			10.00 - 11.30	II	308,312	KEPERAWATAN ANAK I	DE
8	Kamis	11-Jul-19	08.00 - 09.30	II	308,312	KEPERAWATAN JIWA I	BC
			10.00 - 11.30	II	308,312	KEPERAWATAN MATERNITAS	AB
9	Senin	29-Jul-19	08.00 - 09.30	III	308,312	KEPERAWATAN MATERNITAS II	BD
			10.00 - 11.30	III	308,312	KEPERAWATAN KOMUNITAS II	AE
10	Selasa	30-Jul-19	08.00 - 09.30	III	308,312	KEPERAWATAN KELUARGA	BC
			10.00 - 11.30	III	308,312	KEPERAWATAN GERONTIK	DE
11	Rabu	31-Jul-19	08.00 - 09.30	III	308,312	METODOLOGI PENELITIAN	

h. Contoh Jadwal UAS Prodi D3 Keperawatan

Tabel 3. 2 Tabel Jadwal UAS Prodi D3 Keperawatan

JADWAL ASSESSMENT AKHIR SEMESTER D3 KEPERAWATAN							
SEMESTER GENAP TAHUN AJARAN 2018/2019							
NO	HARI	TANGGAL	WAKTU	TK	RUANG	MATA KULIAH	PAWAS
1	Senin	1-Jul-19	08.00 - 09.15	I	304,305	Komunikasi	AC
			09.45 - 11.00	I	304,305	Konsep Wound Care	BE
2	Selasa	2-Jul-19	08.00 - 09.30	I	304,305	Kewirausahaan	CE
			10.00 - 11.30	I	304,305	Dokumentasi Kep.	AC
3	Rabu	3-Jul-19	08.00 - 09.30	I	304,305	Patofisiologi	EA
			10.00 - 11.30	I	304,305	Bahasa Inggris 1	BC
4	Kamis	4-Jul-19	08.00 - 09.30	I	304,305	Farmakologi	DC
			10.00 - 11.30	I	304,305	Etika Keperawatan	AE
5	Jumat	8-Jul-19	08.00 - 09.30	I	304,305	keperawatan dasar 2	DC

i. Lama Hari Pelaksanaan Ujian

Pelaksanaan UAS adalah 2 minggu pada 6 hari efektif sesuai kalender akademik.

Tabel 3. 3 Tabel Lama Hari Pelaksanaan Ujian

No	Hari
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat
6	Sabtu
7	Senin
8	Selasa
9	Rabu
10	Kamis
11	Jumat

12	Sabtu
----	-------

j. Waktu Pelaksanaan UAS Prodi Keperawatan S1

Lama waktu pelaksanaan UAS pada Prodi Keperawatan S1 adalah 90 menit tiap sesi, ujian dilakukan sebanyak 3 sesi dengan jeda waktu istirahat 30 menit pada sesi 1 dengan sesi 2, sedangkan jeda istirahat pada sesi 2 dengan sesi 3 adalah 90 menit.

Tabel 3. 4 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan S1

SESI	WAKTU
1	08.00 – 09.30
2	10.00 – 11.30
3	13.00 – 14.30

k. Waktu Pelaksanaan UAS Prodi Keperawatan D3

Lama waktu pelaksanaan UAS pada Prodi Keperawatan D3 adalah 75 menit tiap sesi, ujian dilakukan sebanyak 3 sesi dengan jeda waktu istirahat 30 menit pada sesi 1 dengan sesi 2, sedangkan jeda istirahat pada sesi 2 dengan sesi 3 adalah 120 menit.

Tabel 3. 5 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan D3

SESI	WAKTU
1	08.00 – 09.15
2	09.45 – 11.00
3	13.00 – 14.15

1. Daftar Mata Kuliah Prodi Keperawatan (D-3)

Tabel 3. 6 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan D3

Program Studi : Keperawatan (D-3)						
Tahun Akademik : 2019/2020						
Semester : Gasal						
NO	Kode MK	Nama Mata Kuliah	SKS	Dosen	Mhs	Semester
1	KPT0601202	Psikologi	2		47	1
2	KPT0601203	Konsep Dasar Keperawatan	2		47	1
3	KPT0601204	Gizi Dan Diit	2		47	1
4	KPT0601206	Antropologi Kesehatan	2		48	1
5	KPT0601401	Ilmu Biomedik Dasar	4		47	1
6	KPT0601505	Keperawatan Dasar	5		47	1
7	NAS0601203	Bahasa Indonesia	2		47	1
8	NAS0601304	Pancasila dan Kewarganegaraan	3		47	1
9	UMM0601008	Ibadah Praktis-Baca Tulis Alquran	0		47	1
10	UMM0601009	Komputer (Extra)	0		47	1
11	UMM0601010	Bahasa Inggris (extra)	0		47	1
12	UMM0601201	Aik I	2		47	1
13	KPT0601215	Metodologi Keperawatan	2		51	3
14	KPT0601216	Manajemen patient Safety	2		52	3
15	KPT0601220	Manajemen Keperawatan	2		51	3
16	KPT0601221	Bahasa Arab	2		51	3
17	KPT0601222	Basic Wound Care	2		51	3
18	KPT0601223	Bahasa Inggris 2	2		51	3

19	KPT0601317	Keperawatan Medikal Bedah I	3		51	3
20	KPT0601318	Keperawatan Keluarga	3		51	3
21	KPT0601319	Keperawatan Maternitas	3		51	3
22	UMM0601203	AIK 3	2		51	3
23	KPT0601232	Prakter Klinik KMB I	2		67	5
24	KPT0601233	Praktek Klinik KMB II	2		66	5
25	KPT0601234	Praktek Klinik Kep Anak	2		67	5
26	KPT0601235	Praktek Klinik Kep Maternitas	2		67	5
27	KPT0601236	Prakterk Klinik Kep Gadar	2		67	5
28	KPT0601340	Karya Tulis Ilmiah	3		6	5

m. Daftar Mata Kuliah Prodi keperawatan (S-1)

Tabel 3. 7 Tabel Waktu Pelaksanaan UAS Prodi Keperawatan S1

NO	Kode MK	Nama Mata Kuliah	SKS	Dosen	Mhs	Semester
1	KPT0603301	Falsafah dan Teori Keperawatan	3		30	1
2	KPT0603302	Konsep Dasar Keperawatan 1	3		30	1
3	KPT0603303	Keperawatan Dasar 1	3		31	1
4	KPT0603404	Ilmu Dasar Keperawatan 1	4		32	1
5	NAS0603201	Pancasila	2		30	1
6	NAS0603203	Bahasa Indonesia	2		30	1
7	NAS0603204	Agama	2		30	1
8	UMM0603008	Ibadah Praktis	0		31	1
9	UMM06030201	Al Islam Kemuhammadiyah 1	2		32	1
10	UMM0603206	Bahasa Inggris Keperawatan 1	2		30	1

NO	Kode MK	Nama Mata Kuliah	SKS	Dosen	Mhs	Semester
11	KPT0603208	Bahasa Inggris Keperawatan 2	2		1	1
12	KPT0603211	Sistem Informasi Keperawatan	2		48	3
13	KPT0603214	Keselamatan Pasien dan keselamatan Kesehatan Kerja dalam Keperawatan	2		48	3
14	KPT0603216	Keperawatan HIV – AIDS	2		48	3
15	KPT0603315	Komunikasi Dalam Keperawatan II	3		48	3
16	KPT0603412	Keperawatan Medikal Bedah I	4		48	3
17	KPT0603413	Keperawatan Mternitas I	4		48	3
18	UMM0605203	Al Islam Kemuhammadiyah III	2		48	3
19	KPT0603325	Keperawatan Medikal Bedah III	3		57	5
20	KPT0603326	Keperawatan Anak II	3		57	5
21	KPT0603327	Keperawatan Komunitas I	3		57	5
22	KPT0603328	Keperawatan Menjelang Ajal dan Paliatif	3		57	5
23	KPT0603427	Keperawatan Kesehatan Jiwa II	4		57	5
24	KPT0603430	Keperawatan Kritis	4		57	5
25	KPT0603234	Biostatistik	2		49	7
26	KPT0603235	Keperawatan Bencana	2		49	7
27	KPT0603438	Manajemen Keperawatan	4			7

B. Analisis Sistem

Analisa system dilakukan untuk menggali informasi tentang penjadwalan ujian semester yaitu menentukan hari , pengawas ujian, ruangan dan mahasiswa pada waktu yang sama ,untuk menentukan batasan- batasan tersebut dapat menggunakan cara yang efektif dalam menyelesaikan permasalahan yaitu dengan merancang sebuah system informasi.

1. Analisis Sistem Yang Sedang Berjalan

Analisis prosedur pada sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, sehingga kelebihan dan kekurangan sistem yang sedang berjalan dapat diketahui. Sistem penjadwalan ujian semester yang sedang berjalan di Fakultas Ilmu Kesehatan yaitu:

- a. Staff TU cek data daftar matakuliah.
- b. TU membuat jadwal dengan menggunakan Microsoft Excel dengan memasukkan hari, sesi, matakuliah, ruangan, pengawas.

1) Matakuliah yang diujikan

Dalam ujian akhir semester ini ada berapa jumlah mata kuliah setiap tingkatan (angkatan)

- a) Tingkat 1 = a, b, c, d
- b) Tingkat 2 = e, f, g, h
- c) Tingkat 3 = i, j, k, l

2) Sesi

Penentuan jadwal ujian dalam satu hari bisa dibuat beberapa sesi, ini dilihat dari jumlah mata kuliah yang ada. Di Prodi Keperawatan S1 dan D3 dibuat satu hari terdapat tiga sesi waktu ujian.

3) Hari

Hari penyelenggaraan ujian dibuat seefektif dan seefisien mungkin, dengan batasan range maksimalnya 2 minggu. Waktu 2 minggu ini adalah hari efektif masa ujian sesuai yang ditentukan berdasarkan kalender akademik yaitu hari Senin sampai dengan hari Sabtu.

4) Waktu

Penyelenggaraan ujian setiap sesi mata kuliah dapat ditentukan sesuai kebijakan masing-masing prodi. Apabila Kaprodi memberikan kebijakan waktu ujian 90 menit per mata kuliah, dengan 3 sesi waktu ujian ditambah waktu istirahat 15 menit, maka bisa dihitung waktu ujian sebagai berikut:

- a) Sesi 1 : 08.00 - 09.30
- b) Sesi 2 : 09.45 - 11.15
- c) Sesi 3 : 13:00 – 14.30

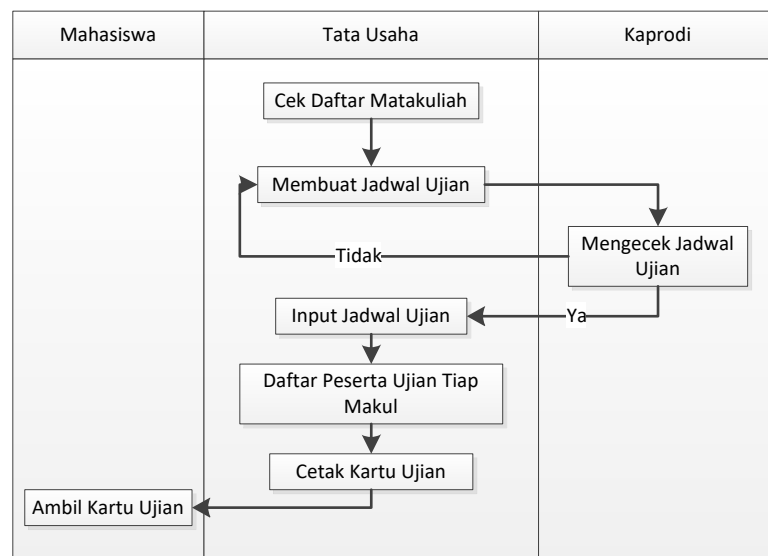
5) Aturan penjadwalan terdiri dari:

- a) Ujian sehari maksimal 5 sesi.
- b) Ruangan dan kelas tidak boleh tabrakan.
- c) Pengawas tidak boleh mengawasi secara berurutan.
- d) Pengawas tidak boleh mengawasi dalam sesi yang sama dan berbeda ruang.
- e) Rungan ujian tidak boleh digunakan untuk ujian dalam sesi yang sama dan berbeda mata kuliah.

6) Ruang Ujian

Ruangan yang digunakan untuk tempat ujian adalah ruang 305 sampai 312. Satu ruangan memiliki kapasitas untuk 30 orang peserta ujian.

- c. TU memberikan jadwal kepada Kaprodi Untuk dikoreksi.
- d. Kaprodi mengkoreski jadwal apabila ada yang perlu dirubah atau diralat.
- e. Jika terjadi kesalah jadwal akan dikembalikan ke TU untuk pembuatan jadwal ulang jika sudah sesuai jadwal akan di cetak oleh TU.
- f. TU mencetak kartu ujian untuk setiap mahasiswa.
- g. Mahasiswa mengambil kartu ujian di ruang TU.



Gambar 3. 4 Flowchart sistem berjalan

Aktor yang ada dalam lingkup system yang sedang berjalan adalah

- a. Kaprodi, yang dilakukan adalah :
 - 1) Mengkoreski jadwal apabila ada yang perlu dirubah atau diralat.
 - 2) Jika terjadi kesalah jadwal akan dikembalikan ke TU untuk pembuatan jadwal ulang.
 - 3) Menyerahkan jadwal yang sudah benar kepada TU.
- b. Staff TU, yang dilakukan adalah :
 - 1) Mengecek daftar matakuliah
 - 2) Membuat jadwal dengan Microsoft Excel dan menentukan pengawas ujian semester.
 - 3) Sekaligus menentukan hari, jam, dan ruangan ujian.

- 4) Memberikan jadwal yang sudah dibuat kepada kaprodi.
 - 5) Jika jadwal sudah benar maka akan di cetak kartu Ujian jika belum akan dibuat ulang jadwal Ujian.
 - 6) Membagikan Kartu Ujian
- c. Mahasiswa
Mengambil kartu ujian di TU.

2. Bagian yang crucial dan potensi bermasalah :

- a. Pengawas tidak boleh mengawasi dalam sesi yang sama dan berbeda ruang dalam satu sesi. Dalam hal ini terjadi tabrak pengawas dari jadwal ujian di keperawatan S1 dan D3 yaitu pada hari yang sama waktu yang sama ruang berbeda dengan pengawas yang sama.

Tabel 3. 8 Tabel Contoh Pengawas yang mengawasi dalam sesi yang sama dan berbeda ruang dalam satu sesi.

Jadwal UAS Keperawatabn D3							
No	Hari	Tanggal	Waktu	TK	Ruang	Matakuliah	Pengawas
1	Senin	1-Jul-19	09.45 – 11.00	1	304,305	Konsep wound care	B,E
Jadwal UAS Keperawatabn S1							
No	Hari	Tanggal	Waktu	TK	Ruang	Matakuliah	Pengawas
1	Senin	1-Jul-19	10.00 – 11.30	1	308,312	Ilmu dasar keperawatan II	B,D

- b. Ruangan dan kelas tidak boleh bertabrakan.

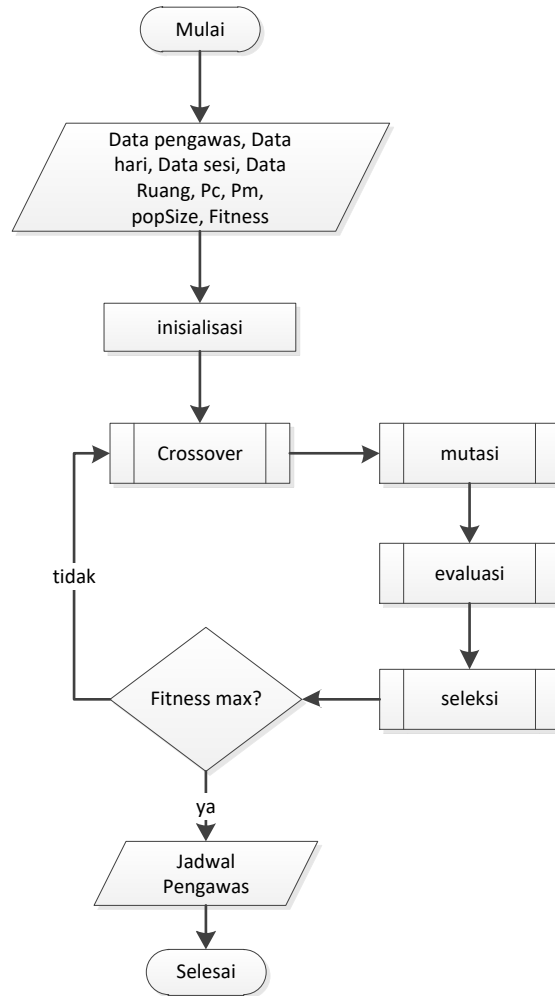
Dalam hal ini tidak mungkin terjadi tabrakan ruang ujian dikarenakan jumlah ruangan yang mencukupi. Dikatakan mencukupi karena pada jadwal yang sudah berjalan hanya butuh 4 ruang,

sedangkan ruangan yang tersedia ada 8 ruang yaitu Ruang 305, Ruang 306, Ruang 307, Ruang 308, Ruang 309, Ruang 310, Ruang 311, Ruang 312.

Selama ini dalam penjadwalan ujian semester fakultas ilmu kesehatan masih manual dan membutuhkan waktu yang lama dalam pembuatannya juga rumit. Sehingga hal ini berdampak pada kurang efektif dalam pembuatan jadwal ujian semester dan kurang efisien. dan belum ada sistem informasi penjadwalan ujian semester berbasis web.

3. Flowchart Siklus Penyelesaian Masalah

Siklus penyelesaian masalah dengan algoritma genetika diawali dengan pengkodean kromosom yang dilanjutkan dengan inisialisasi populasi awal, reproduksi, evaluasi dan proses seleksi untuk menentukan solusi optimal. Berikut ini pada gambar 3.1 ditunjukkan flowchart siklus penyelesaian masalah penjadwalan pengajar pada Fakultas Ilmu Kesehatan.



Gambar 3. 5 Gambar Flowchart Siklus Pemecahan Masalah dengan Algoritma Genetika

4. Siklus Penyelesaian Masalah menggunakan Algoritma Genetika

Parameter-parameter yang digunakan pada proses algoritma genetika diantaranya adalah: (Tusty, Arina, Endah, Wisnu, Bayu, 2016)

- a. Jumlah individu dalam setiap populasi (popsize)
Jumlah individu dalam setiap populasi ada 5 individu kromosom.
- b. Jumlah generasi
Jumlah generasi bergantung pada jumlah iterasi, iterasi akan berhenti ketika nilai fitness mencapai nilai optimal.

c. Probabilitas crossover (p_c)

Nilai probabilitas crossover merupakan nilai yang menentukan jumlah offspring hasil proses crossover, pada penelitian ini nilai probabilitas crossover yang digunakan yaitu 0.4.

Nilai probabilitas crossover yang digunakan yaitu 0.4.

d. Probabilitas mutasi (p_m)

Nilai probabilitas mutasi merupakan nilai yang menentukan jumlah offspring hasil proses mutasi. Pada penelitian ini nilai probabilitas mutasi yang digunakan yaitu 0.2.

5. Inisialisasi Populasi Awal

Pada penelitian ini, inisialisasi populasi awal dilakukan secara random sesuai dengan representasi kromosom yang dipilih sebanyak popsize yang diinginkan. Pada tahap ini dilakukan penentuan representasi kromosom yang digunakan untuk populasi. Representasi yang digunakan dalam penelitian ini adalah representasi permutasi berbasis integer. Bilangan integer yang digunakan dalam representasi kromosom merupakan bilangan hasil dari pengkodean data pengajar yang dimasukkan dan disesuaikan dengan hari, sesi dan kelas yang ada berikut ini ditunjukkan nilai pengkodean data pengajar, kelas, dan sesi pengajaran perhari.

Tabel 3. 9 Pengkodean Pegawai

Pengawas	Code
Ira Budi	P001
Prasetyo	P002
Nurul Niawati	P003
Suwarji Wibowo	P004
Guno Ariyanto	P005

Tabel 3. 10 Tabel Pengkodean Jam

WAKTU	CODE
07.00 – 08.30	Jam ke-1
08.45 – 10.15	Jam ke-2
10.30 – 12.00	Jam ke-3
13.00 – 14.30	Jam ke-4
14.45 – 16.15	Jam ke-5

Tabel 3. 11 Tabel Pengkodean Waktu

Tanggal	Jam	Code
Senin, 27 Januari 2020	07.00 – 08.30	Sesi 1
Senin, 27 Januari 2020	08.45 – 10.15	Sesi 2
Senin, 27 Januari 2020	10.30 – 12.00	Sesi 3
Senin, 27 Januari 2020	13.00 – 14.30	Sesi 4
Senin, 27 Januari 2020	14.45 – 16.15	Sesi 5

Tabel 3. 12 Tabel Pengkodean Ruang

RUANG	CODE
305	R1

306	R2
307	R3
308	R4
309	R5
310	R4
312	R5
313	R6

Tabel 3. 13 Tabel Pengkodean Makul Keperawatan S1

MATAKULIAH	SEMESTER	CODE
Falsafah dan Teori Keperawatan	1	MKS1
Konsep Dasar Keperawatan 1	1	MKS2
Keperawatan Dasar 1	1	MKS3
Ilmu Dasar Keperawatan 1	1	MKS4
Pancasila	1	MKS5
Bahasa Indonesia	1	MKS6
Agama	1	MKS7
Ibadah Praktis	1	MKS8
Al Islam Kemuhammadiyah 1	1	MKS9

Bahasa Inggris Keperawatan 1	1	MKS10
Bahasa Inggris Keperawatan 2	1	MKS11
Sistem Informasi Keperawatan	3	MKS12
Keselamatan Pasien dan keselamatan Kesehatan Kerja dalam Keperawatan	3	MKS13
Keperawatan HIV - AIDS	3	MKS14
Komunikasi Dalam Keperawatan II	3	MKS15
Keperawatan Medikal Bedah I	3	MKS16
Keperawatan Mternitas I	3	MKS17
Al Islam Kemuhammadiyah III	3	MKS18
Keperawatan Medikal Bedah III	5	MKS19
Keperawatan Anak II	5	MKS20
Keperawatan Komunitas I	5	MKS21
Keperawatan Menjelang Ajal dan Paliatif	5	MKS22
Keperawatan Kesehatan Jiwa II	5	MKS23
Keperawatan Kritis	5	MKS24
Biostatistik	7	MKS25
Keperawatan Bencana	7	MKS26
Manajemen Keperawatan	7	MKS27

Tabel 3. 14 Tabel Pengkodean Makul Keperawatan D3

Nama Mata Kuliah	Semester	CODE
Psikologi	1	MKD1
Konsep Dasar Keperawatan	1	MKD2
Gizi Dan Diit	1	MKD3
Antropologi Kesehatan	1	MKD4
Ilmu Biomedik Dasar	1	MKD5
Keperawatan Dasar	1	MKD6
Bahasa Indonesia	1	MKD7
Pancasila dan Kewarganegaraan	1	MKD8
Ibadah Praktis-Baca Tulis Alquran	1	MKD9
Komputer (Extra)	1	MKD10
Bahasa Inggris (extra)	1	MKD11
Aik I	1	MKD12
Metodologi Keperawatan	3	MKD13
Manajemen patient Safety	3	MKD14
Manajemen Keperawatan	3	MKD15
Bahasa Arab	3	MKD16
Basic Wound Care	3	MKD17
Bahasa Inggris 2	3	MKD18
Keperawatan Medikal Bedah I	3	MKD19
Keperawatan Keluarga	3	MKD20
Keperawatan Maternitas	3	MKD21
AIK 3	3	MKD22

Nilai pengkodean pengawas, ruang, waktu, hari, matakuliah yang telah dibuat dapat membantu menentukan representasi kromosom dari permasalahan. Kromosom yang dibentuk merupakan susunan gen yang mewakili pengawas dengan jumlah gen kromosom yaitu sebanyak 10 (1 hari x 5 sesi x 2 ruang). Berikut ini ditunjukkan hasil representasi kromosom pada tabel

Tabel 3. 15 Tabel Representasi Kromosom Pengawas Hari ke 1

INDIVIDU	Hari 1											
	Sesi 1				Sesi 2				Sesi 3			
	MKS1		MKD1		MKS2		MKD2		MKS3		MKD3	
	R1	R2	R3	R4	R5	R6	R7	R8	R1	R2	R3	R4
1	P1	P5	P3	P3	P2	P4	P1	P4	P5	P5	P1	P1
2	P2	P1	P2	P5	P1	P4	P3	P2	P3	P3	P4	P1
3	P5	P3	P1	P4	P2	P3	P3	P1	P5	P3	P1	P5
4	P2	P3	P1	P2	P4	P2	P4	P5	P3	P1	P2	P5
5	P3	P3	P4	P2	P5	P4	P3	P1	P2	P4	P5	P1

1. Reproduksi

Pada tahap reproduksi terdapat 2 metode yang digunakan yaitu crossover dan mutasi. Metode yang digunakan untuk crossover adalah one point cut crossover dan untuk mutasi menggunakan metode reciprocal exchange mutation.

4.1 Crossover

Crossover merupakan proses tukar silang antara 2 individu. Metode yang digunakan di sini adalah one point cut crossover. Metode one point cut crossover memisahkan 1 kromosom menjadi dua bagian dengan cara menentukan 1 titik secara random pada 1 kromosom. Berikut ini merupakan ilustrasi perhitungan crossover pada permasalahan optimal pengawas UAS prodi keperawatan S1 dan D3 FIKES UMMgl.

$$P_c = 0,4; \text{popSize} = 5$$

$$\begin{aligned} \text{Jumlah offspring yang dihasilkan pada proses crossover} &= P_c * \text{popSize} \\ &= 0,4 * 5 \end{aligned}$$

= 2

Menentukan parent secara random dari populasi pada tabel 3.15 dan menentukan secara random titik potong crossover.

Parent1 = individu ke -3

P5	P3	P1	P4	P2	P3	P3	P1	P5	P3	P1	P5
----	----	----	----	----	----	----	----	----	----	----	----

Parent 2 = individu ke -5

P3	P3	P4	P2	P5	P4	P3	P1	P2	P4	P5	P1
----	----	----	----	----	----	----	----	----	----	----	----

Berdasarkan titik potong yang telah ditentukan, dilakukan proses pertukaran posisi dari parent 1 dan parent 2. Berikut ini adalah hasil dari one cut point crossover dari parent 1 dan parent 2.

Child 1

P5	P3	P1	P4	P2	P3	P3	P1	P2	P4	P5	P1
----	----	----	----	----	----	----	----	----	----	----	----

Child 2

P3	P3	P4	P2	P5	P4	P3	P1	P5	P3	P1	P5
----	----	----	----	----	----	----	----	----	----	----	----

4.2 Mutasi

Metode mutasi yang digunakan adalah reciprocal exchange mutation. Metode ini bekerja dengan memilih dua posisi (exchange point / xp) secara random kemudian menukar nilai pada posisi tersebut. Berikut ini merupakan ilustrasi proses mutasi.

$P_m = 0,2$; popSize = 5

Offspring yang dihasilkan pada proses mutase = $P_m * \text{popSize}$

$$= 0,2 * 5$$

$$= 1$$

Menentukan kromosom yang akan dilakukan proses mutasi secara random dari populasi pada Tabel 4:

Misal menggunakan individu ke-4 dengan xp1 dan xp2 yang ditentukan secara random secara berturut-turut pada gen posisi ke-3 untuk xp1 dan posisin ke-9.

Parent 3 : Individu ke-4

P2	P3	P1	P2	P4	P2	P4	P5	P3	P1	P2	P5
----	----	----	----	----	----	----	----	----	----	----	----

Cild 3 :

P2	P3	P3	P2	P4	P2	P4	P5	P1	P1	P2	P5
----	----	----	----	----	----	----	----	----	----	----	----

4.3 Evaluasi

Proses evaluasi digunakan untuk menghitung nilai fitness pada setiap kromosom. Semakin besar kromosom semakin baik kromosom tersebut dijadikan calon solusi yang optimal.

Dari proses inialisasi dan reproduksi, didapatkan kumpulan individu sebagai berikut:

Tabel 3. 16 Tabel Proses Evaluasi

Cild	Hari 1											
	Sesi 1				Sesi 2				Sesi 3			
	MKS1	MKD1	MKS2	MKD2	MKS3	MKD3						
1	P5	P3	P1	P4	P2	P3	P3	P1	P2	P4	P5	P1
2	P3	P3	P4	P2	P5	P4	P3	P1	P5	P3	P1	P5
3	P2	P3	P3	P2	P4	P2	P4	P5	P1	P1	P2	P5

Karena masalah ini merupakan penjadwalan dengan pinalti yaitu bentrok, maka nilai fitness dapat dihitung dengan persamaan berikut

$$f(x) = \frac{1}{1 + \sum pinalti}$$

Tabel 3. 17 Tabel Perhitungan Nilai Fitness pada child (offspring)

Child	Hari 1												Bentrok	Fitness
	Sesi 1				Sesi 2				Sesi 3					
	MKS1		MKD1		MKS2		MKD2		MKS3		MKD3			
1	P5	P3	P1	P4	P2	P3	P3	P1	P2	P4	P5	P1	2	0.333333
2	P3	P3	P4	P2	P5	P4	P3	P1	P5	P3	P1	P5	4	0.2
3	P2	P3	P3	P2	P4	P2	P4	P5	P1	P1	P2	P5	8	0.111111

Dari individu dsan offspring yang terpilih berdasarkan nilai fitness, maka menjadi populasi untuk generasi berikutnya.

4.4 Seleksi

Proses seleksi digunakan untuk memilih individu (parent) dari populasi child (offspring) yang bertahan hidup pada generasi selanjutnya semakit besar nilai fitness semakin besar peluang untuk terplih. Seleksi ini dilakukan agar generasi berikutnya memiliki populasi yang lebih baik daripada generasi selanjutnya. Dengan menggunakan elitism selection, dipilih popSize individu terbaik dari kumpulan individu dari populasi dan offspring. Berdasarkan data pada tabel 9 berikut:

Tabel 3. 18 Tabel Kumpulan Individu & offspring pada hari 1

INDIVIDU	Hari 1												Bentrok	Fitnes
	Sesi 1				Sesi 2				Sesi 3					
	MKS1		MKD1		MKS2		MKD2		MKS3		MKD3			
	R1	R2	R3	R4	R5	R6	R7	R8	R1	R2	R3	R4		
1	P1	P5	P3	P3	P2	P4	P1	P4	P5	P5	P1	P1	6	0.142857
2	P2	P1	P2	P5	P1	P4	P3	P2	P3	P3	P4	P1	2	0.333333
3	P5	P3	P1	P4	P2	P3	P3	P1	P5	P3	P1	P5	4	0.2
4	P2	P3	P1	P2	P4	P2	P4	P5	P3	P1	P2	P5	4	0.2
5	P3	P3	P4	P2	P5	P4	P3	P1	P2	P4	P5	P6	2	0.333333
6	P5	P3	P1	P4	P2	P3	P3	P1	P2	P4	P5	P6	2	0.333333
7	P3	P3	P4	P2	P5	P4	P3	P1	P5	P3	P1	P5	4	0.2
8	P2	P3	P3	P2	P4	P2	P4	P5	P1	P1	P2	P5	8	0.111111

Setelah dilakukan proses seleksi, individu yang terpilih seleksi pada tabel 3.19 berikut:

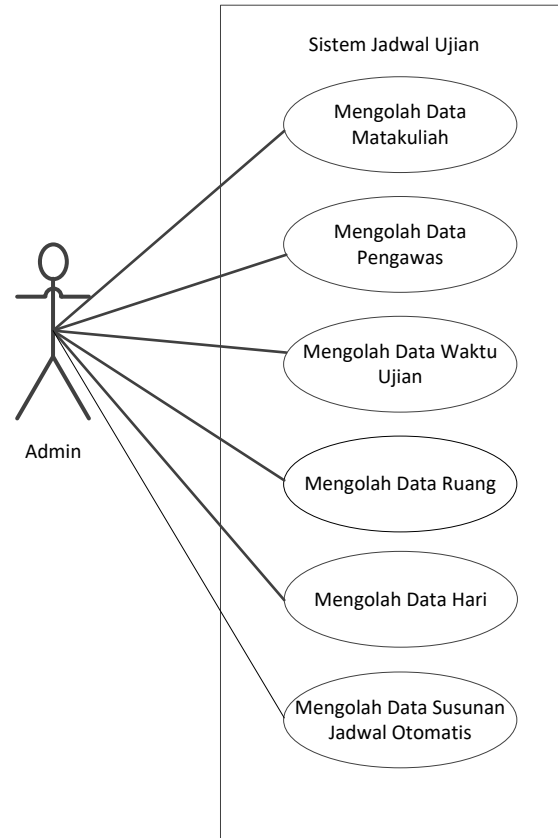
Tabel 3. 19 Tabel Individu Terpilih pada Generasi Berikutnya

Individu	Asal	Hari 1												Bentrok	Fitnes
		Sesi 1				Sesi 2				Sesi 3					
		MKS1		MKD1		MKS2		MKD2		MKS3		MKD3			
		R1	R2	R3	R4	R5	R6	R7	R8	R1	R2	R3	R4		
1	2	P2	P1	P2	P5	P1	P4	P3	P2	P3	P3	P4	P1	2	0.333333
2	5	P3	P3	P4	P2	P5	P4	P3	P1	P2	P4	P5	P1	2	0.333333
3	6	P5	P3	P1	P4	P2	P3	P3	P1	P2	P4	P5	P1	2	0.333333
4	3	P5	P3	P1	P4	P2	P3	P3	P1	P5	P3	P1	P5	4	0.2
5	4	P2	P3	P1	P2	P4	P2	P4	P5	P3	P1	P2	P5	4	0.2

C. Perancangan Unified Modeling Language Diagram

UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak (Windu Gata dan Grace, 2013).

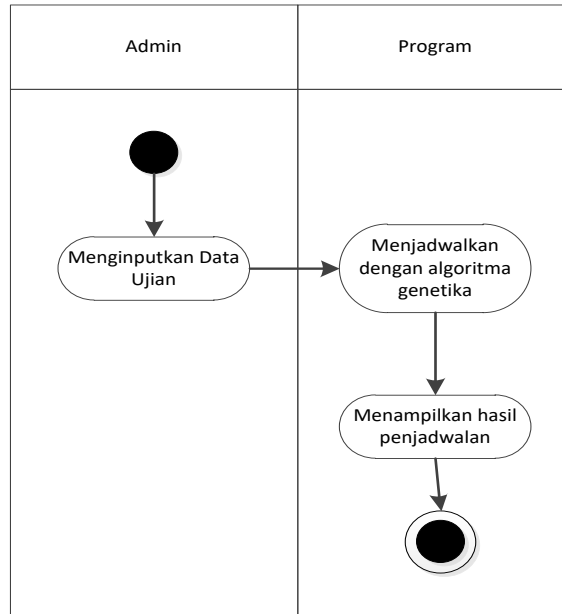
1. Usecase Diagram



Gambar 3. 6 Use Case Diagram

Gambar 3.5 menjelaskan bahwa didalam aplikasi terdapat satu aktor, yaitu admin yang bertugas untuk mengelola data matakuliah, ruangan, pengawas, waktu ujian, dan jadwal ujian.

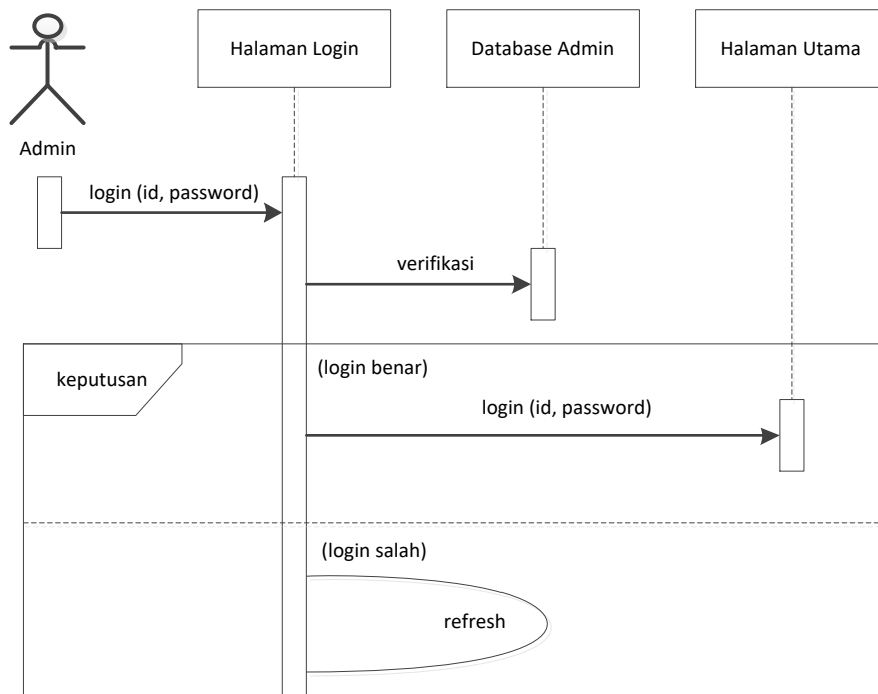
2. Activity Diagram



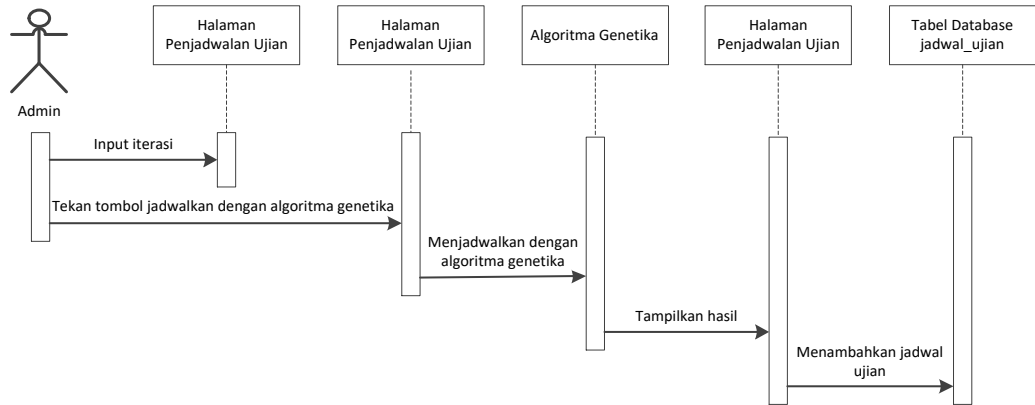
Gambar 3. 7 Activity Diagram

Dari gambar 3.6 dapat dilihat bahwa admin dapat menjadwalkan ujian dengan menggunakan program algoritma genetika.

3. Sequence Diagram

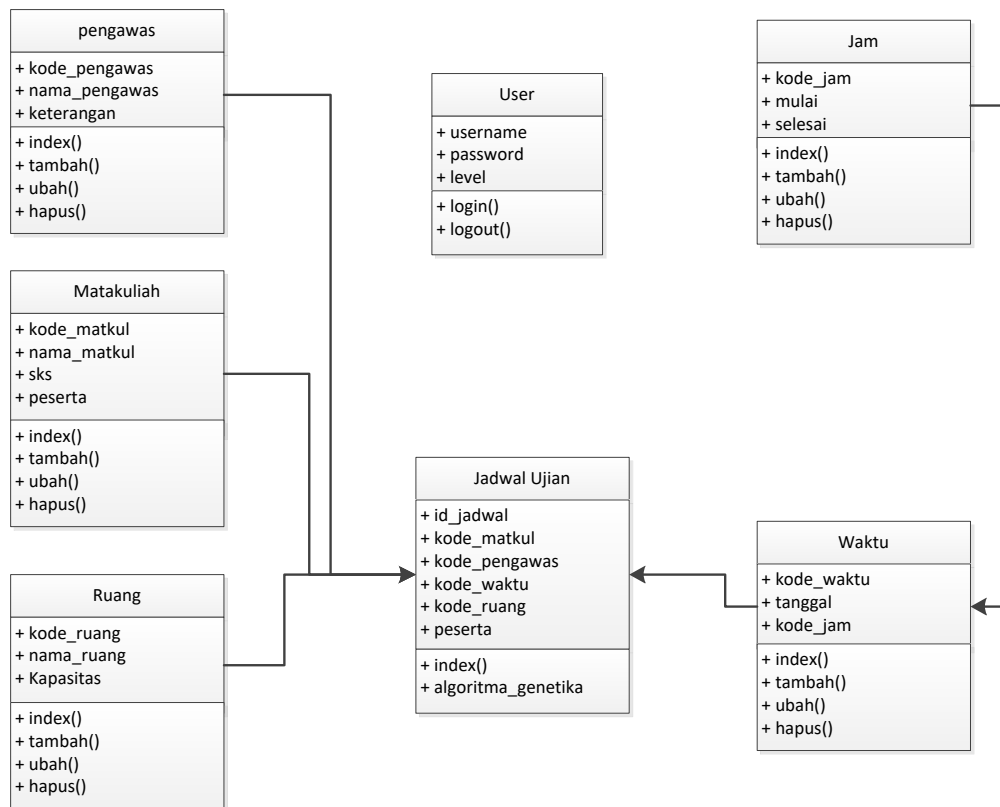


Gambar 3. 8 Sequence Diagram Login



Gambar 3. 9 Sequence Diagram Penjadwalan Ujian dengan Algoritma Genetika

4. Class Diagram



Gambar 3. 10 Class Diagram

Dari gambar 3.9 kita dapat melihat terdapat 7 class yang terdiri dari class user yang berdiri sendiri, class waktu, class pengawas, class

jam, class ruang dan class makul yang terhubung dengan class jadwal ujian.

D. Perancangan Basis Data

Struktur database yang dijelaskan entitas dan atribut nya sebagai berikut

1. Menentukan Entitas dan Atribut.

Tabel 3. 20 Tabel Entitas

No.	Entitas
1	Admin
2	Hari
3	Mata Kuliah
4	Pengawas
5	Ruangan
6	Waktu Ujian
7	Tabel Jadwal

Terdapat 7 entitas dipilih pada sistem ini yaitu entitas Admin, Hari, Mata Kuliah, Pengawas, Ruangan, Waktu Ujian, Tabel Jadwal. Setiap entitas memiliki atribut yang dijelaskan pada tabel berikut.

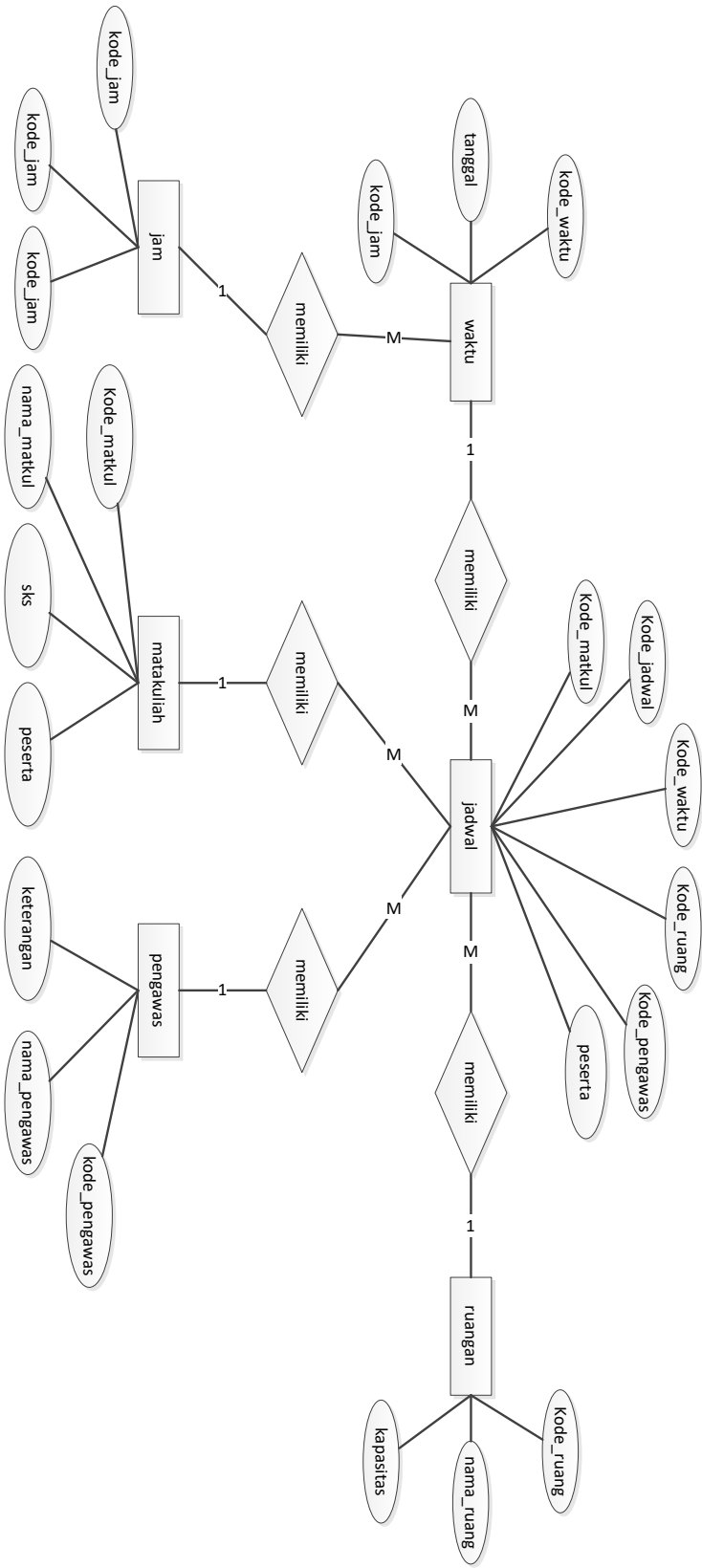
Tabel 3. 21 Tabel Penentuan Entitas dan Atribut

NO	Entitas	Atribut
1	User	id_user*
		Username
		Password
		Level

2	Mata Kuliah	kode_matkul*
		nama_matkul
		Sks
		Peserta
3	Pengawas	kode_pengawas*
		nama_pengawas
		Keterangan
4	Ruangan	kode_ruang*
		nama_ruang
		Kapasitas
5	Waktu	id_waktu*
		Tanggal
		kode_jam
6	Jam	kode_jam
		Mulai
		Selesai
7	Jadwal	kode_jadwal*
		kode_waktu**
		kode_ruang**
		kode_matkul**
		kode_pengawas**
		Peserta

Pada Tabel 3.21 Primary Key ditunjukkan dengan lambing bintang (*) dan Foreign key ditunjukkan dengan lambing du bintang (**). Setelah

mentukan entitas dan masing-masing atributnya yang digunakan dalam sistem maka akan diperoleh relasi antara entitas yang dapat digambarkan dalam ERD.



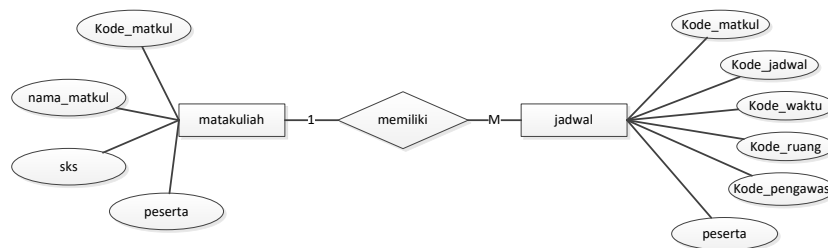
Gambar 3. 11 Gambar ERD

Penjelasan Gambar 3.11 Gambar ERD sistem yang diusulkan dimana dapat dilihat terdapat 6 entitas dan masing – masing memiliki atribut. Setiap entitas dalam sistem berhubungan dengan entitas lain.

2. Relasi dan Kardinalitas

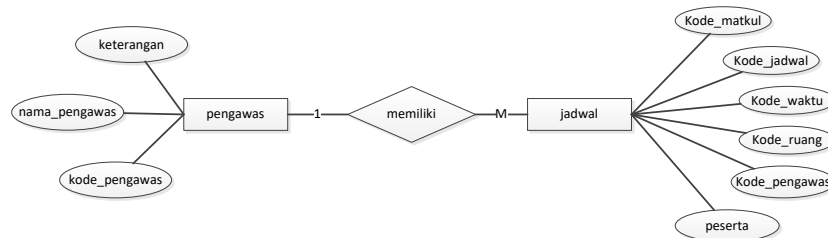
Dengan melihat hubungan entitas, maka dapat ditentukan derajat keanggotaan dan kelas keanggotaan masing-masing entitas yang dilihat pada gambar dibawah ini

a. Relasi Matakuliah dan Jadwal



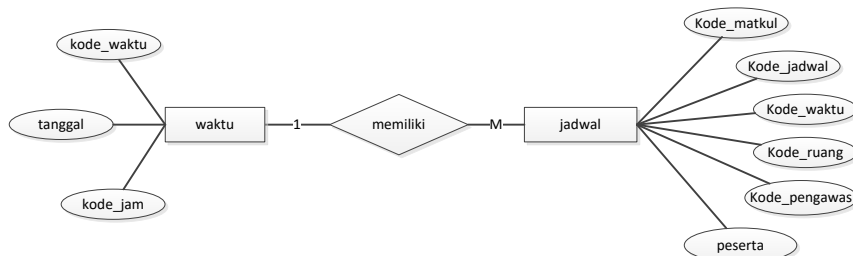
Gambar 3. 12 Relasi Matakuliah dan Jadwal

b. Relasi Pengawas dan Jadwal



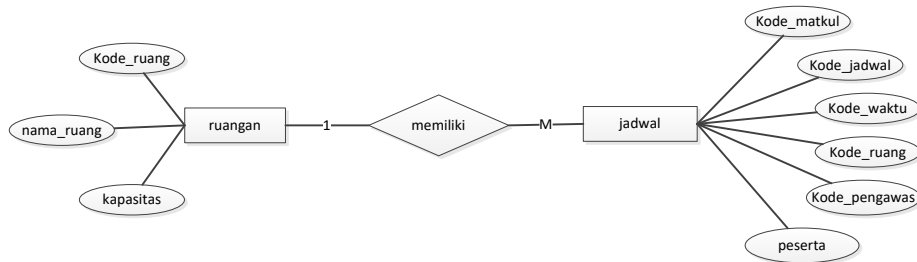
Gambar 3. 13 Relasi Matakuliah dan Jadwal

c. Relasi Waktu Ujian dan Jadwal



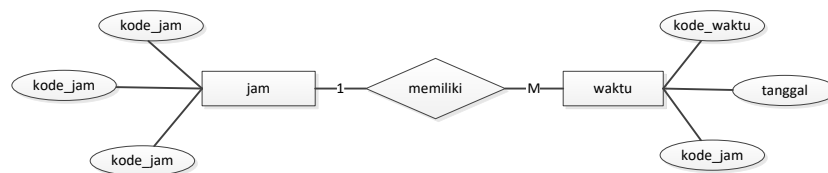
Gambar 3. 14 Relasi Waktu Ujian dan Jadwal

d. Relasi Ruangan dan Jadwal



Gambar 3. 15 Relasi Ruangan dan Jadwal

e. Relasi Jam dan Waktu



Gambar 3. 16 Relasi Jam dan Waktu

E. Desain Database

Struktur database yang dijelaskan secara rinci atribut dan penjelasannya.

Struktur data tersebut sebagai berikut

1. Rancangan Tabel Matakuliah

Rancangan tabel ini digunakan untuk menyimpan data mata kuliah ke dalam basis data. Adapun yang menjadi primary key di tabel ini adalah kode_matkul dan rancangan tabel mata kuliah dapat dilihat pada tabel berikut :

Tabel 3. 22 Tabel Rancangan Tabel Matakuliah

Nama Field	Tipe Data	Panjang	Keterangan
kode_matkul	varchar	16	Kode Mata Kuliah (Primary Key)
nama_matkul	varchar	255	Nama Matakuliah
Sks	int	11	Jumlah Sks
Peserta	int	11	Jumlah peserta dalam satu matakuliah

2. Rancangan Tabel Pengawas

Rancangan tabel ini digunakan untuk menyimpan data pengawas ke dalam basis data. Adapun yang menjadi primary key di tabel ini adalah kode_pengawas dan rancangan tabel pengawas dapat dilihat pada tabel berikut :

Tabel 3. 23 Tabel Rancangan Tabel Pengawas

Nama Field	Tipe Data	Panjang	Keterangan
kode_pengawas	varchar	11	Kode Pengawas (Primary Key)
nama_pengawas	varchar	30	Nama Pengawas
keterangan	varchar	50	Jabatan Pengawas

3. Rancangan Tabel Ruangan

Rancangan tabel ini digunakan untuk menyimpan data ruangan ke dalam basis data. Adapun yang menjadi primary key di tabel ini adalah kode_ruang dan rancangan tabel ruangan dapat dilihat pada tabel berikut:

Tabel 3. 24 Tabel Rancangan Tabel Ruangan

Nama Field	Tipe Data	Panjang	Keterangan
kode_ruang	int	11	Kode ruang (Primary Key)
nama_ruang	varchar	30	Nama Ruangan
Kapasitas	int	11	Kapasitas Ruangan

4. Rancangan Tabel Waktu Ujian

Rancangan tabel ini digunakan untuk menyimpan data waktu ujian ke dalam basis data. Adapun yang menjadi primary key di tabel ini adalah kode_waktu dan rancangan tabel waktu ujian dapat dilihat pada tabel berikut:

Tabel 3. 25 Tabel Waktu Ujian

Nama Field	Tipe Data	Panjang	Keterangan
kode_waktu	int	11	Kode Waktu (Primary Key)
Tanggal	date	-	Jam Akhir
kode_jam	varchar	20	Kode Jam (Foreign Key)

5. Rancangan Tabel Jadwal

Rancangan tabel ini digunakan untuk menyimpan data jadwal ke dalam basis data. Adapun yang menjadi primary key di tabel ini adalah kode_jadwal dan rancangan tabel jadwal dapat dilihat pada tabel berikut :

Tabel 3. 26 Tabel Rancangan Tabel Jadwal

Nama Field	Tipe Data	Panjang	Keterangan
kode_jadwal	int	11	Kode Jadwal (Primary Key)
kode_matkul	varchar	16	Kode Matakuliah (Foreign Key)
kode_pengawas	varchar	16	Kode Pengawas (Foreign Key)
kode_waktu	int	11	Kode Waktu (Foreign Key)
kode_ruang	int	11	Kode Ruang (Foreign Key)
Peserta	Int	11	Jumlah Peserta Ujian

F. Perancangan Interface

Berikut untuk tampilan rancangan aplikasi yang akan diterapkan pada aplikasis sebagai berikut :

1. Tampilan Halaman Login

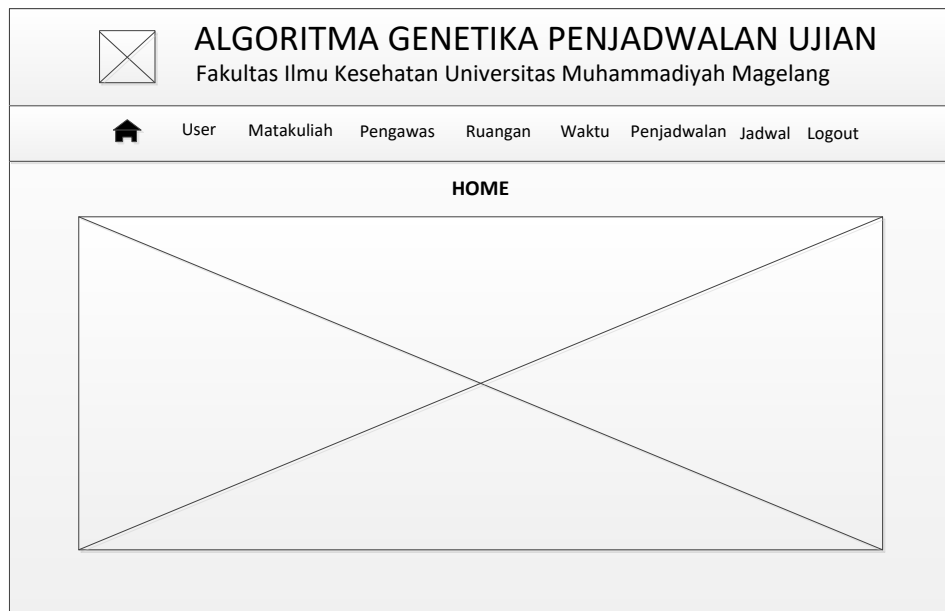


The image shows a login form with a light gray background. At the top center, the word "Login" is written in a large, bold, black font. Below it, there are two input fields: "Username" and "Password", each with a corresponding text box. At the bottom center, there is a rounded rectangular button labeled "Masuk".

Gambar 3.15 Tampilan Halaman Login

Tampilan ini digunakan untuk memberikan keamanan data yang berhubungan dengan penjadwalan sidang. Sebelum memasuki sistem petugas harus memasukkan *username* dan kata *password* atau kata sandi terlebih dahulu.

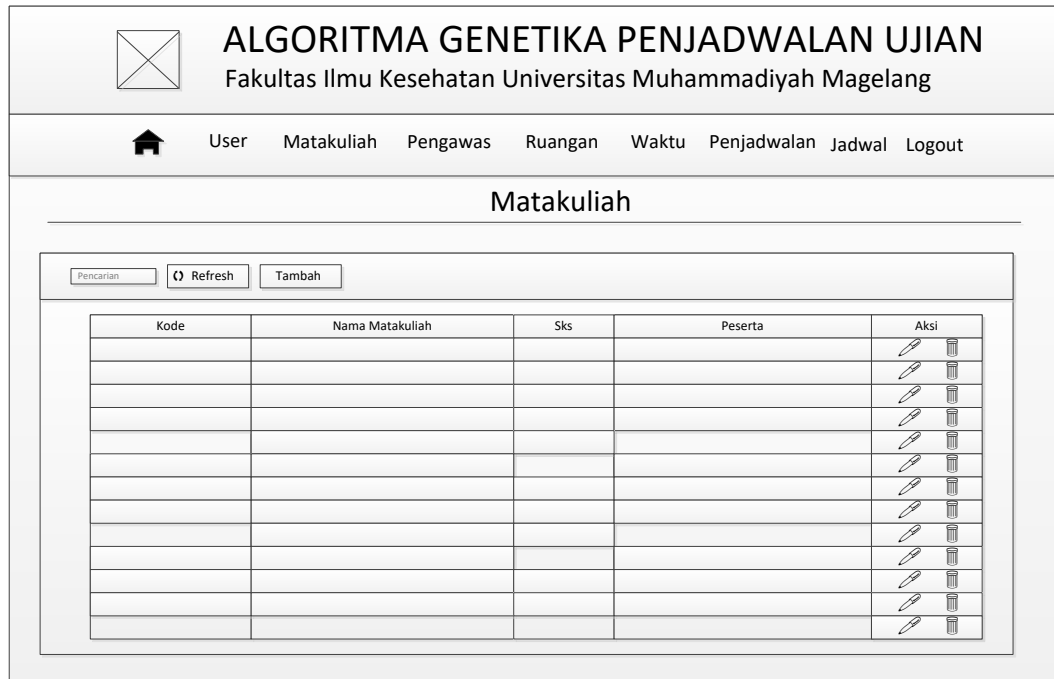
2. Tampilan Halaman Menu



The image shows a menu page with a light gray background. At the top left, there is a square icon with an 'X' inside. To its right, the text "ALGORITMA GENETIKA PENJADWALAN UJIAN" is displayed in a bold, black font, followed by "Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang" in a smaller font. Below this, there is a horizontal navigation bar with a home icon and the following menu items: "User", "Matakuliah", "Pengawas", "Ruangan", "Waktu", "Penjadwalan", "Jadwal", and "Logout". Below the navigation bar, the word "HOME" is centered. The main content area is a large rectangle with a diagonal 'X' across it, indicating a placeholder for content.

Gambar 3.16 Tampilan Halaman Menu

4. Tampilan Halaman Matakuliah




















ALGORITMA GENETIKA PENJADWALAN UJIAN
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

User Matakuliah Pengawas Ruangan Waktu Penjadwalan Jadwal Logout

Matakuliah

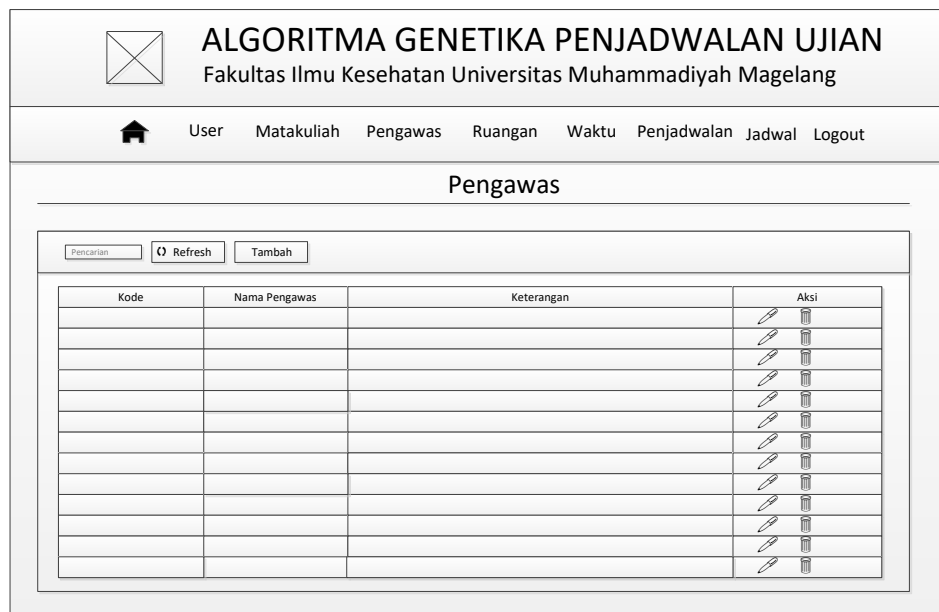
Pencarian Refresh Tambah

Kode	Nama Matakuliah	Sks	Peserta	Aksi
				 
				 
				 
				 
				 
				 
				 
				 
				 
				 

Gambar 3. 18 Gambar Tampilan Halaman Matakuliah

Tampilan Halaman Matakuliah adalah antarmuka yang digunakan untuk menampilkan data mata kuliah di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang.

5. Tampilan Halaman Pengawas





















ALGORITMA GENETIKA PENJADWALAN UJIAN
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

User Matakuliah Pengawas Ruangan Waktu Penjadwalan Jadwal Logout

Pengawas

Pencarian Refresh Tambah

Kode	Nama Pengawas	Keterangan	Aksi
			 
			 
			 
			 
			 
			 
			 
			 
			 

Gambar 3. 19 Gambar Halaman Pengawas

BAB IV

IMPLEMENTASI DAN PENGUJIAN

A. IMPLEMENTASI SISTEM

Implementasi sistem merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis dalam Bahasa yang dapat dimengerti oleh mesin computer yaitu diimplementasikan dalam Bahasa pemrograman. Hal ini diharapkan agar penerapan perangkat lunak siap untuk dioperasikan sesuai dengan semestinya dan dapat memenuhi kebutuhan penggunanya.

1. Implementasi perangkat Pembuat aplikasi

Sistem penjadwalan ujian akhir semester menggunakan metode algoritma genetika dibuat menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) yaitu :

Tabel 4. 1 Data *Hardware*

NO.	ITEM	SPESIFIKASI
1	Prosesor	Intel® Core™ i3-3110M Processor (2.4 GHz, Cache 3MB)
2	RAM	6 GB
3	HDD/Hardisk	500 GB Serial ATA 5400 RPM
4	Monitor	14" WXGA 1366 x 768 BrightView LED-backlit
5	Mouse, Keyboard	Standart
6	Printer	Dot Matrix / Ink Jet

Tabel 4. 2 Data *Software*

No.	ITEM
1	Windows 7 Ultimate
2	Sublime Text 3
3	Xampp
4	Web Browser
5	Mysql

2. Implementasi Database

a. Implentasi Tabel User

Tabel user digunakan untuk menyimpan username dan password yang digunakan untuk akses masuk kedalam sistem, Implementasi tabel user dapat dilihat pada gambar 4.1.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	ID	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	user	varchar(16)	latin1_swedish_ci		Ya	Tidak ada		
3	pass	varchar(16)	latin1_swedish_ci		Ya	Tidak ada		
4	level	varchar(16)	latin1_swedish_ci		Ya	Tidak ada		

Gambar 4. 1 gambar implementasi tabel user.

b. Implementasi Tabel Matakuliah

Tabel matakuliah digunakan untuk menyimpan biodata matakuliah yang diujikan di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Implementasi tabel matakuliah dapat dilihat pada gambar 4.2.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	kode_matkul	varchar(16)	latin1_swedish_ci		Tidak	Tidak ada		
2	nama_matkul	varchar(255)	latin1_swedish_ci		Ya	Tidak ada		
3	sks	int(11)			Ya	Tidak ada		
4	peserta	int(11)			Ya	Tidak ada		

Gambar 4. 2 gambar implementasi tabel matakuliah.

c. Implementasi Tabel Waktu

Tabel hari digunakan untuk menyimpan hari, tanggal dan jam yang dapat digunakan untuk kegiatan penjadwalan ujian di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Implementasi tabel hari dapat dilihat pada gambar 4.3.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id_waktu	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	tanggal	date		Ya		Tidak ada		
3	kode_jam	varchar(255)	latin1_swedish_ci		Ya	0		

Gambar 4. 3 gambar implementasi tabel waktu.

d. Implementasi Tabel Sesi

Tabel sesi digunakan untuk menyimpan nama sesi, waktu mulai dan waktu selesai ujian pada Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Implementasi tabel hari dapat dilihat pada gambar 4.4.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	kode_jam	varchar(16)	latin1_swedish_ci		Tidak	Tidak ada		
2	mulai	time		Ya		Tidak ada		
3	selesai	time		Ya		Tidak ada		

Gambar 4. 4 gambar implementasi tabel sesi.

e. Implementasi Tabel Ruang

Tabel ruang digunakan untuk menyimpan nama ruang dan kapasitas ruang untuk ujian Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Implementasi tabel ruang dapat dilihat pada gambar 4.5.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	kode_ruang	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	nama_ruang	varchar(255)	latin1_swedish_ci		Ya	Tidak ada		
3	kapasitas	int(11)		Ya		Tidak ada		

Gambar 4. 5 gambar implementasi tabel ruang.

f. Implementasi Tabel Pengawas

Tabel pengawas digunakan untuk menyimpan nama pengawas yang mengawasi ujian di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Implementasi tabel pengawas dapat dilihat pada gambar 4.6.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	kode_pengawas	varchar(256)	latin1_swedish_ci		Tidak	Tidak ada		
2	nama_pengawas	varchar(256)	latin1_swedish_ci		Ya	Tidak ada		
3	keterangan	varchar(256)	latin1_swedish_ci		Ya			

Gambar 4. 6 gambar implementasi tabel pengawas.

g. Implementasi Tabel Jadwal Ujian Akhir Semester

Tabel Jadwal Ujian Akhir Semester adalah tabel yang digunakan untuk menyimpan jadwal ujian akhir semester hasil dari perhitungan Algoritma genetika. Implementasi tabel jadwal kuliah dapat dilihat pada gambar 4.7.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
1	id_jadwal	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	matkul	varchar(16)	latin1_swedish_ci		Ya	Tidak ada		
3	pengawas	varchar(16)	latin1_swedish_ci		Ya	Tidak ada		
4	waktu	int(11)			Ya	Tidak ada		
5	ruang	int(11)			Ya	Tidak ada		
6	peserta	int(11)			Ya	Tidak ada		

Gambar 4. 7 gambar implementasi tabel jadwal ujian akhir semester.

3. Implementasi Antarmuka

a. Implementasi Antarmuka Login Admin

The screenshot shows a web browser window with the URL `localhost/127.0.0.1/egujian-ci/index.php/user/tambah`. The page title is "Algoritma Genetika Penjadwalan Ujian" and the subtitle is "Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang". The main content area is titled "Tambah User" and contains the following form elements:

- Username ***: A text input field.
- Password ***: A text input field.
- Level**: A dropdown menu with "Administrator" selected.
- Buttons**: "Simpan" (Save) and "Kembali" (Back).

The footer of the page contains the copyright information: "© 2020 Algoritma Genetika Penjadwalan Ujian Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang".

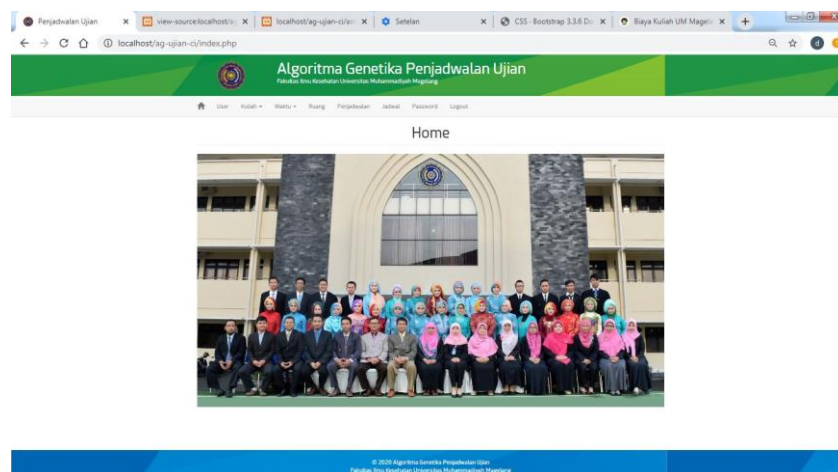
Gambar 4. 8 gambar login admin.

Seperti pada gambar 4.8 adalah sebuah aplikasi login Admin Member yang memberikan fungsi keamanan terhadap data yang ada di dalam sistem tersebut. Untuk membuka login, user

harus memasukkan kata sandi sampai kata sandi yang sudah ditentukan benar.

b. Implementasi Antarmukan Beranda

Antarmuka beranda adalah antarmuka yang pertama kali setelah login muncul saat pengguna masuk kedalam sistem. Antarmuka ini dapat dilihat pada gambar 4.9.




Gambar 4. 9 Implementasi Antarmuka Beranda

c. Implementasi Antarmuka Matakuliah

1) Implementasi Antarmuka Tampilan Data Matakuliah

Dalam Interface ini komponen data matakuliah dari prodi S1 Keperawatan dan D3 keperawatan dimasukkan menjadi 1 agar dapat dihitung menggunakan algoritma genetika. matakuliah dari 2 prodi ini digunakan untuk membentuk gen dan menyusun kromosom dalam pembangkitan awal sebelum jadwal ujian semester dapat diproses

Implementasi antarmuka tampilan data mata kuliah adalah antarmuka yang digunakan untuk menampilkan data mata kuliah di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Tampilan ini dapat dilihat pada gambar 4.10.

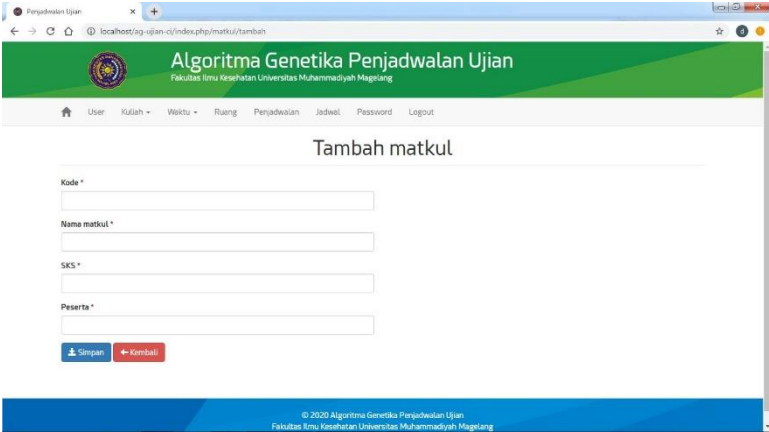


Kode	Nama Matkul	SKS	Peserta	Aksi
KPT0501202	Psikologi	2	47	[Edit] [Hapus]
KPT0501203	Konsep Dasar Keperawatan	2	47	[Edit] [Hapus]
KPT0501204	Gizi Dan Diet	2	47	[Edit] [Hapus]
KPT0501206	Antropologi Kesehatan	2	48	[Edit] [Hapus]
KPT0501215	Metodologi Keperawatan	2	51	[Edit] [Hapus]
KPT0501216	Manajemen patient Safety	2	52	[Edit] [Hapus]
KPT0501220	Manajemen Keperawatan	2	51	[Edit] [Hapus]
KPT0501221	Bahasa Arab	2	51	[Edit] [Hapus]
KPT0501222	Basic Wound Care	2	51	[Edit] [Hapus]

Gambar 4. 10 Implementasi Antarmuka Data Matakuliah

2) Implementasi Antarmuka Tambah Data Mata Kuliah

Implementasi antarmuka tambah data mata kuliah adalah antarmuka yang digunakan untuk menambah data mata kuliah di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Tampilan ini dapat dilihat pada gambar 4.11.



Tambah matkul

Kode *

Nama matkul *

SKS *

Peserta *

[Simpan] [Kembali]

© 2020 Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

Gambar 4. 11 Implementasi Antarmuka Tambah Matakuliah.

3) Implementasi Antarmuka Ubah Data Mata Kuliah

Implementasi antarmuka tampilan ubah data mata kuliah adalah antarmuka yang digunakan untuk mengubah data mata kuliah di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Tampilan ini dapat dilihat pada gambar 4.12.

Ubah matkul

Kode
KPT0501202

Nama*
Psikologi

SKS*
2

Peserta*
47

Simpan Kembali

© 2020 Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

Gambar 4. 12 Implementasi Antarmuka Ubah Matakuliah

d. Implementasi Antarmuka Pengawas

1) Implementasi Antarmuka Data Pengawas

Implementasi antarmuka data pengawas adalah tampilan yang digunakan untuk menampilkan data pengawas ujian. Tampilan antarmuka pengawas dapat dilihat pada gambar 4.13.

Pengawas

Kode	Nama Dosen	Keterangan	Aksi
P001	Ira Budi Setyowidani, SE	Kopila Tata Usaha	[Edit] [Hapus]
P002	Prasetyo Budi Utama, S.Kom	Staff	[Edit] [Hapus]
P003	Nursi Nawati	Staff	[Edit] [Hapus]
P004	Suryawati Wibowo	Staff	[Edit] [Hapus]
P005	Guno Ariyanto	Staff	[Edit] [Hapus]

© 2020 Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

Gambar 4. 13 Implementasi Antarmuka Data Pengawas

2) Implementasi Antarmuka Tambah Data Pengawas

Implementasi antarmuka tambah data pengawas adalah tampilan yang digunakan untuk menambahkan data pengawas ujian. Tampilan ini dapat dilihat pada gambar 4.14.

Gambar 4. 14 Implementasi Antarmuka Tambah Data Pengawas.

3) Implementasi Antarmuka Ubah Data Pengawas

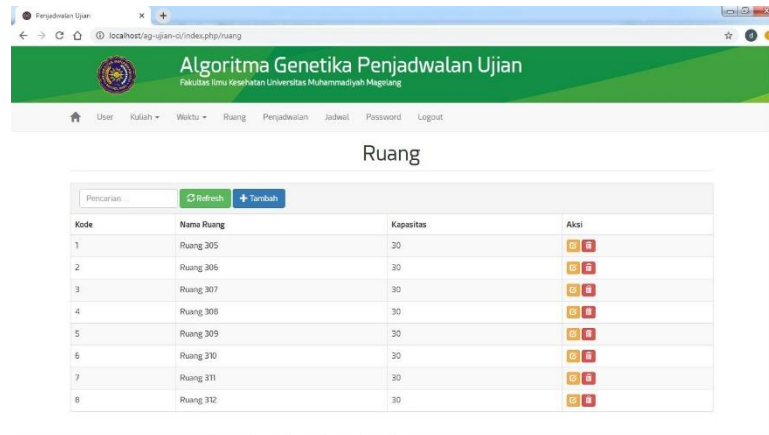
Implementasi antarmuka tampilan ubah data pengawas adalah tampilan yang digunakan untuk mengubah data pengawas ujian. Tampilan ini dapat dilihat pada gambar 4.15.

Gambar 4. 15 Implementasi Antarmuka Ubah Data Pengawas

e. Implementasi Antarmuka Ruang

1) Implementasi Antarmuka Tampilan Data Ruang

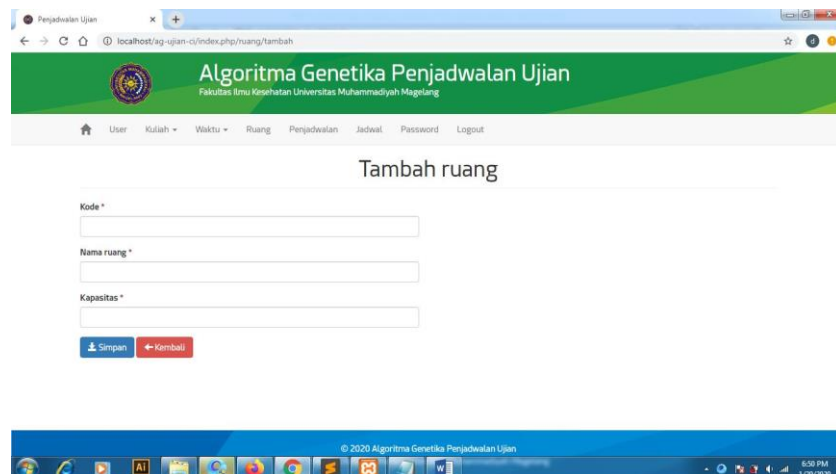
Implementasi antarmuka data ruang adalah halaman yang digunakan untuk menampilkan data ruang. Tampilan ini dapat dilihat pada gambar 4.16 .



Gambar 4. 16 Implementasi Antarmuka Tampilan Data Ruang

2) Implementasi Antarmuka Tambah Data Ruang

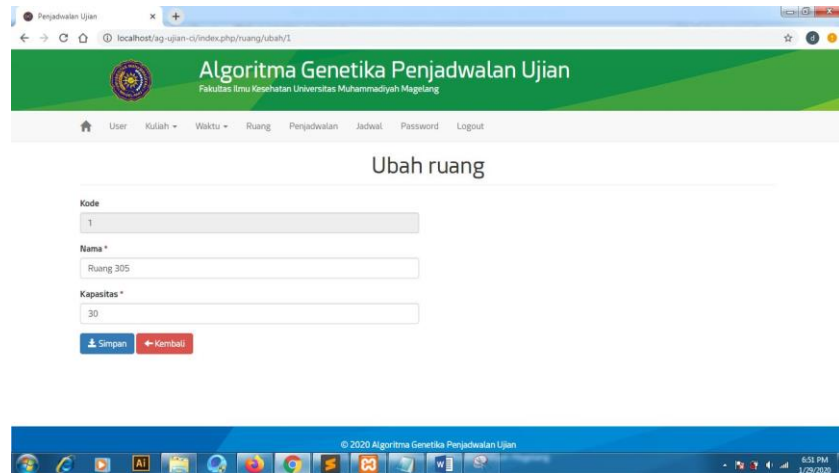
Implementasi antarmuka tambah data ruang adalah halaman yang digunakan untuk menambah data ruang. Tampilan ini dapat dilihat pada gambar 4.17.



Gambar 4. 17 Implementasi Antarmuka Tambah Data Ruang

3) Implementasi Antarmuka Ubah Data Ruang

Implementasi antarmuka ubah data ruang adalah halaman yang digunakan untuk mengubah data ruang. Tampilan ini dapat dilihat pada gambar 4.18.



Gambar 4. 18 Implementasi Antarmuka Ubah Data Ruang

f. Implementasi Antarmuka Jam

1) Implementasi Antarmuka Tampilan Data Jam

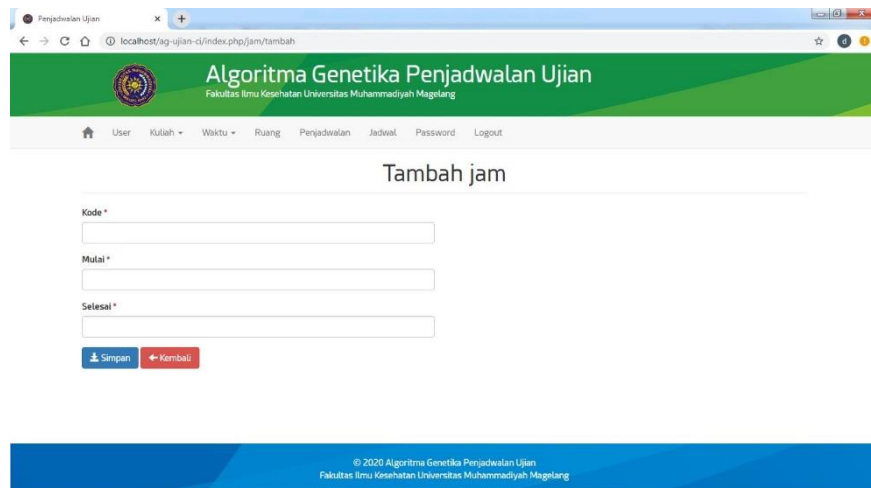
Implementasi antarmuka data sesi adalah halaman yang digunakan untuk menampilkan data jam ujian. Tampilan ini dapat dilihat pada gambar 4.19.



Gambar 4. 19 Implementasi Antarmuka Input Data Jam

2) Implementasi Antarmuka Tambah Data Sesi

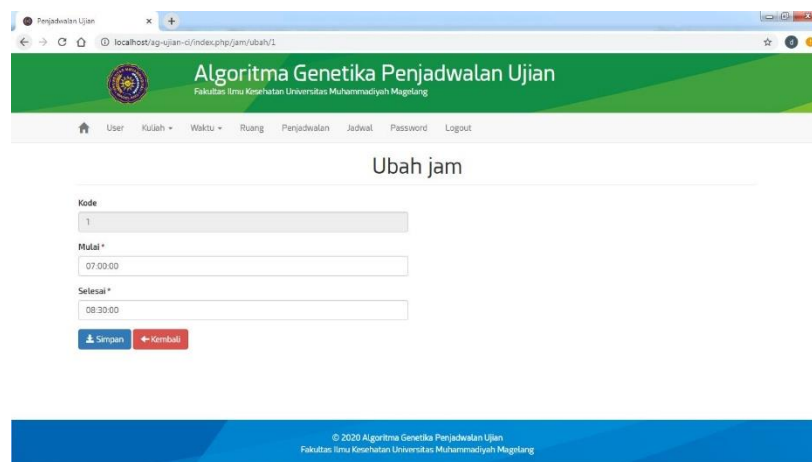
Implementasi antarmuka tambah data sesi adalah halaman yang digunakan untuk menambah data sesi. Tampilan ini dapat dilihat pada gambar 4.20.



Gambar 4. 20 Implementasi Antarmuka Tambah Data Sesi

3) Implementasi Antarmuka Ubah Data Jam

Implementasi antarmuka ubah data sesi adalah halaman yang digunakan untuk mengubah data Jam. Tampilan ini dapat dilihat pada gambar 4.21.

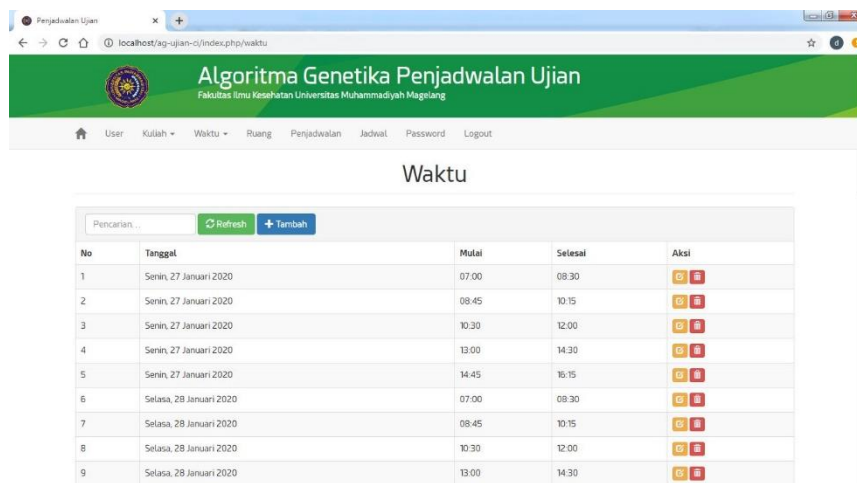


Gambar 4. 21 Implementasi Antarmuka Ubah Data Sesi

g. Implementasi Antarmuka Waktu

1) Implementasi Antarmuka Tampilan Data Hari

Implementasi antarmuka data hari adalah halaman yang digunakan untuk menampilkan data Waktu. Tampilan ini dapat dilihat pada gambar 4.22.

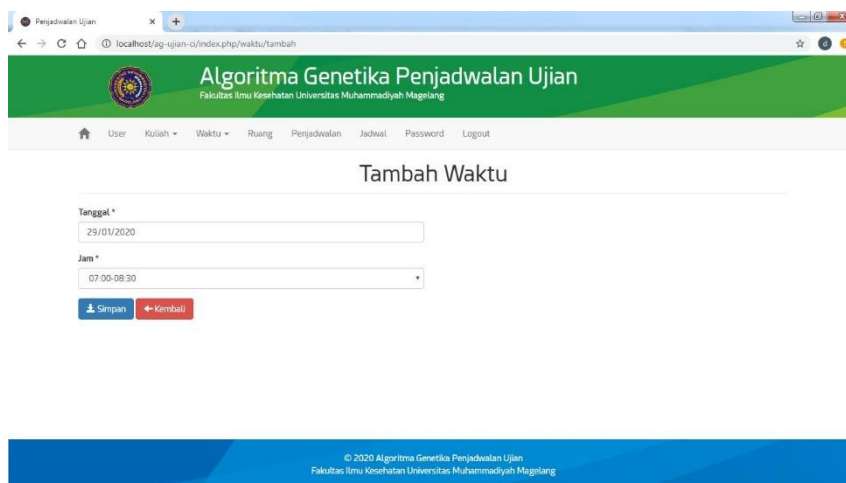


No	Tanggal	Mulai	Selesai	Aksi
1	Senin, 27 Januari 2020	07:00	08:30	[Edit] [Hapus]
2	Senin, 27 Januari 2020	08:45	10:15	[Edit] [Hapus]
3	Senin, 27 Januari 2020	10:30	12:00	[Edit] [Hapus]
4	Senin, 27 Januari 2020	13:00	14:30	[Edit] [Hapus]
5	Senin, 27 Januari 2020	14:45	16:15	[Edit] [Hapus]
6	Selasa, 28 Januari 2020	07:00	08:30	[Edit] [Hapus]
7	Selasa, 28 Januari 2020	08:45	10:15	[Edit] [Hapus]
8	Selasa, 28 Januari 2020	10:30	12:00	[Edit] [Hapus]
9	Selasa, 28 Januari 2020	13:00	14:30	[Edit] [Hapus]

Gambar 4. 22 Implementasi Antarmuka Tampilan Data Waktu

2) Implementasi Antarmuka Tambah Data Waktu

Implementasi antarmuka tambah data waktu adalah halaman yang digunakan untuk menambah data waktu. Tampilan ini dapat dilihat pada gambar 4.23.



Tambah Waktu

Tanggal *
23/01/2020

Jam *
07:00-08:30

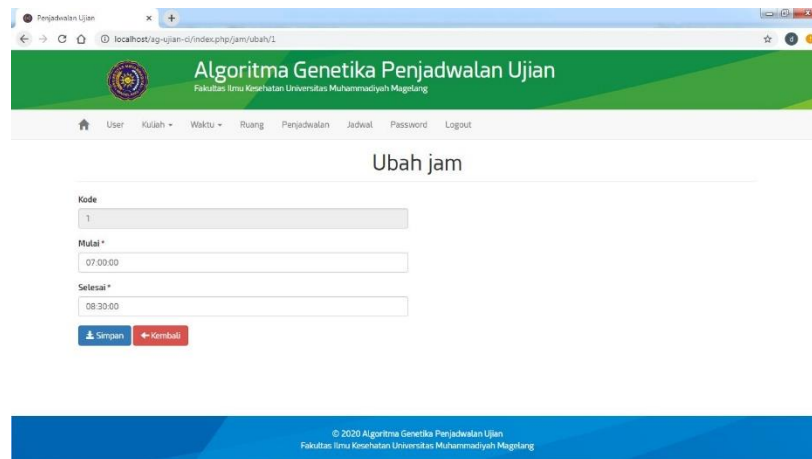
[Simpan] [Kembali]

© 2020 Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

Gambar 4. 23 Implementasi Antarmuka Tambah Data Hari

4) Implementasi Antarmuka Ubah Data Hari

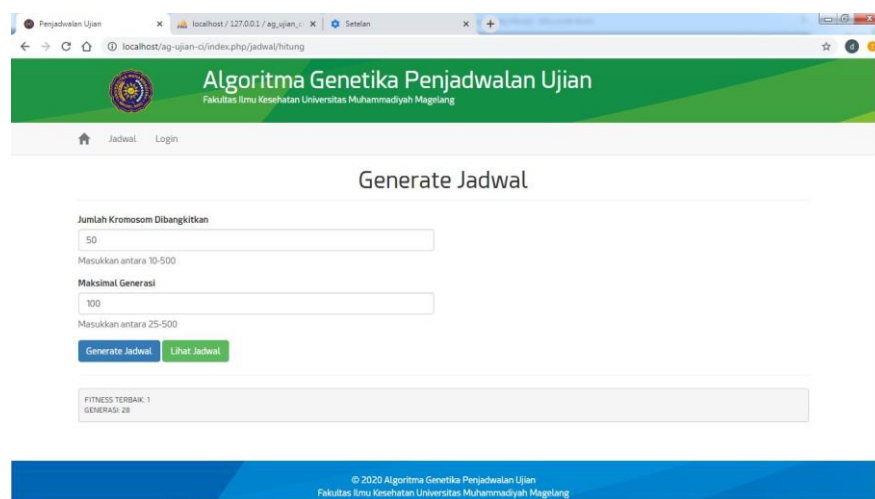
Implementasi antarmuka ubah data hari adalah halaman yang digunakan untuk mengubah data hari. Tampilan ini dapat dilihat pada gambar 4.24.



Gambar 4. 24 Implementasi Antarmuka Ubah Data Hari

h. Implementasi Antarmuka Proses Penjadwalan

Implementasi antarmuka proses penjadwalan adalah antarmuka yang digunakan untuk melakukan proses penjadwalan ujian akhir semester menggunakan algoritma genetika. Tampilan implementasi antarmuka proses penjadwalan dapat dilihat pada gambar 4.25.



Gambar 4. 25 Implementasi Antarmuka Input Data Proses Penjadwalan

4. Script Program

Berikut merupakan *script* program yang dibuat berdasarkan *class diagram* yang telah dirancang

a. Login

```

public function login()
{
    if($this->session->userdata('login'))
    {
        redirect();
    }

    $data['title'] = 'Login';

    $this->form_validation->set_rules('user', 'Username', 'required');
    $this->form_validation->set_rules('pass', 'Password', 'required|callback_ceklogin');

    if ($this->form_validation->run() === FALSE)
    {
        load_view('login', $data);
    }
    else
    {
        $level = $this->input->post('level');
        $user = $this->input->post('user');
        redirect();
    }
}

function ceklogin()
{
    $user = $this->input->post('user');
    $pass = $this->input->post('pass');

    $user_login = $this->user_model->login($user, $pass);

    if($user_login)
    {
        $this->session->set_userdata('ID', $user_login->ID);
        $this->session->set_userdata('login', TRUE);
        $this->session->set_userdata('user', $user_login->user);
        $this->session->set_userdata('level', $user_login->level);

        return true;
    }
    else
    {
        $this->form_validation->set_message('ceklogin', 'Login gagal');
        return false;
    }
}

```

Gambar 4. 26 *script* program cek login

Gambar 4.26 merupakan fungsi validasi user akan masuk ke sistem. Pada proses ini dilakukan cek status user, sehingga user yang masuk ke sistem diarahkan ke halaman sesuai dengan hak aksesnya.

b. Menampilkan Data Matakuliah

```

<?php
class Matkul extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('matkul_model');
    }

    public function index()
    {
        $data['rows'] = $this->matkul_model->tampil($this->input->get('search'));
        $data['title'] = 'Matkul';

        load_view('matkul/index', $data);
    }

    public function tambah()

```

```

{
    $this->form_validation->set_rules('kode', 'Kode', 'required|is_unique[tb_matkul.kode_matkul]');
    $this->form_validation->set_rules('nama', 'Nama', 'required');
    $this->form_validation->set_rules('sks', 'SKS', 'required|is_natural_no_zero');
    $this->form_validation->set_rules('peserta', 'Peserta', 'required');

    $data['title'] = 'Tambah matkul';

    if ($this->form_validation->run() === FALSE)
    {
        load_view('matkul/tambah', $data);
    }
    else
    {
        $fields = array(
            'kode_matkul' => $this->input->post('kode'),
            'nama_matkul' => $this->input->post('nama'),
            'sks' => $this->input->post('sks'),
            'peserta' => $this->input->post('peserta'),
        );
        $this->matkul_model->tambah($fields);
        redirect('matkul');
    }
}

public function ubah( $ID = null )
{
    $this->form_validation->set_rules('nama', 'Nama', 'required');
    $this->form_validation->set_rules('sks', 'SKS', 'required|is_natural_no_zero');
    $this->form_validation->set_rules('peserta', 'Peserta', 'required');

    $data['title'] = 'Ubah matkul';

    if ($this->form_validation->run() === FALSE)
    {
        $data['row'] = $this->matkul_model->get_matkul($ID);
        load_view('matkul/ubah', $data);
    }
    else
    {
        $fields = array(
            'nama_matkul' => $this->input->post('nama'),
            'sks' => $this->input->post('sks'),
            'peserta' => $this->input->post('peserta'),
        );
        $this->matkul_model->ubah($fields, $ID);
        redirect('matkul');
    }
}

public function hapus( $ID = null )
{
    $this->matkul_model->hapus($ID);
    redirect('matkul');
}
}

```

Gambar 4. 27 *script* Data Matakuliah

Gambar 4.27. merupakan script untuk menampilkan data matakuliah yang sudah pernah di inputkan. Serta dapat menambah data dan mengubah data matakuliah yang sudah ada.

c. *Script* Data Waktu

```

<?php
class Waktu extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('waktu_model');
        $this->load->model('jam_model');
    }

    public function index()
    {
        $data['rows'] = $this->waktu_model->tampil($this->input->get('search'));
        $data['title'] = 'Waktu';

        load_view('waktu/index', $data);
    }
}

```

```

public function tambah()
{
    $this->form_validation->set_rules( 'tanggal', 'Tanggal', 'required' );
    $this->form_validation->set_rules( 'kode_jam', 'Jam', 'required' );

    $data['title'] = 'Tambah Waktu';

    if ( $this->form_validation->run() === FALSE )
    {
        load_view('waktu/tambah', $data);
    }
    else
    {
        $fields = array(
            'tanggal' => $this->input->post('tanggal'),
            'kode_jam' => $this->input->post('kode_jam'),
        );
        $this->waktu_model->tambah($fields);
        redirect('waktu');
    }
}

public function ubah( $ID = null )
{
    $this->form_validation->set_rules( 'tanggal', 'Tanggal', 'required' );
    $this->form_validation->set_rules( 'kode_jam', 'Jam', 'required' );

    $data['title'] = 'Ubah Waktu';

    if ( $this->form_validation->run() === FALSE )
    {
        $data['row'] = $this->waktu_model->get_waktu($ID);
        load_view('waktu/ubah', $data);
    }
    else
    {
        $fields = array(
            'tanggal' => $this->input->post('tanggal'),
            'kode_jam' => $this->input->post('kode_jam'),
        );
        $this->waktu_model->ubah($fields, $ID);
        redirect('waktu');
    }
}

public function hapus( $ID = null )
{
    $this->waktu_model->hapus($ID);
    redirect('waktu');
}

public function cetak( $search = "" )
{
    $data['rows'] = $this->waktu_model->tampil($search);
    load_view_cetak('waktu/cetak', $data);
}
}

```

Gambar 4. 28 *script* Data Waktu

Gambar 4.28. merupakan script untuk menampilkan data Hari yang sudah pernah di inputkan. Serta dapat menambah data dan mengubah data Hari yang sudah ada.

d. *Script Data Jam*

```

<?php
class Jam extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('jam_model');
    }

    public function index()
    {
        $data['rows'] = $this->jam_model->tampil($this->input->get('search'));
        $data['title'] = 'Jam';

        load_view('jam/index', $data);
    }

    public function tambah()
    {
        $this->form_validation->set_rules( 'kode', 'Kode', 'required|is_unique[tb_jam.kode_jam]' );
        $this->form_validation->set_rules( 'mulai', 'Mulai', 'required' );
        $this->form_validation->set_rules( 'selesai', 'Selesai', 'required' );

        $data['title'] = 'Tambah jam';

        if ($this->form_validation->run() === FALSE)
        {
            load_view('jam/tambah', $data);
        }
        else
        {
            $fields = array(
                'kode_jam' => $this->input->post('kode'),
                'mulai' => $this->input->post('mulai'),
                'selesai' => $this->input->post('selesai'),
            );
            $this->jam_model->tambah($fields);
            redirect('jam');
        }
    }

    public function ubah( $ID = null )
    {
        $this->form_validation->set_rules( 'mulai', 'Mulai', 'required' );
        $this->form_validation->set_rules( 'selesai', 'Selesai', 'required' );

        $data['title'] = 'Ubah jam';

        if ($this->form_validation->run() === FALSE)
        {
            $data['row'] = $this->jam_model->get_jam($ID);
            load_view('jam/ubah', $data);
        }
        else
        {
            $fields = array(
                'mulai' => $this->input->post('mulai'),

```

```

        'selesai' => $this->input->post('selesai'),
    );
    $this->jam_model->ubah($fields, $ID);
    redirect('jam');
}
}

public function hapus( $ID = null )
{
    $this->jam_model->hapus($ID);
    redirect('jam');
}
}

```

Gambar 4. 29 *script* Data Jam

Gambar 4.29. merupakan script untuk menampilkan data Jam yang sudah pernah di inputkan. Serta dapat menambah data jam dan mengubah data jam yang sudah ada.

e. Script Data Ruang

```

<?php
class Ruang extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('ruang_model');
    }

    public function index()
    {
        $data['rows'] = $this->ruang_model->tampil($this->input->get('search'));
        $data['title'] = 'Ruang';

        load_view('ruang/index', $data);
    }

    public function tambah()
    {
        $this->form_validation->set_rules( 'kode', 'Kode', 'required|is_unique[tb_ruang.kode_ruang] );
        $this->form_validation->set_rules( 'nama', 'Nama', 'required' );
        $this->form_validation->set_rules( 'kapasitas', 'Kapasitas', 'required' );

        $data['title'] = 'Tambah ruang';

        if ($this->form_validation->run() === FALSE)
        {
            load_view('ruang/tambah', $data);
        }
        else
        {
            $fields = array(
                'kode_ruang' => $this->input->post('kode'),
                'nama_ruang' => $this->input->post('nama'),
                'kapasitas' => $this->input->post('kapasitas'),
            );
            $this->ruang_model->tambah($fields);
            redirect('ruang');
        }
    }

    public function ubah( $ID = null )

```

```

{
    $this->form_validation->set_rules( 'nama', 'Nama', 'required' );
    $this->form_validation->set_rules( 'kapasitas', 'Kapasitas', 'required' );

    $data['title'] = 'Ubah ruang';

    if ($this->form_validation->run() === FALSE)
    {
        $data['row'] = $this->ruang_model->get_ruang($ID);
        load_view('ruang/ubah', $data);
    }
    else
    {
        $fields = array(
            'nama_ruang' => $this->input->post('nama'),
            'kapasitas' => $this->input->post('kapasitas'),
        );
        $this->ruang_model->ubah($fields, $ID);
        redirect('ruang');
    }
}

public function hapus( $ID = null )
{
    $this->ruang_model->hapus($ID);
    redirect('ruang');
}
}

```

Gambar 4. 30 *Script* Data Ruang

Gambar 4.30. merupakan script untuk menampilkan data ruang yang sudah pernah di inputkan. Serta dapat menambah data ruang dan mengubah data ruang yang sudah ada.

f. *Script* Data Pengawas

```

<?php
class Pengawas extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('pengawas_model');
    }

    public function index()
    {
        $data['rows'] = $this->pengawas_model->tampil($this->input->get('search'));
        $data['title'] = 'Pengawas';

        load_view('pengawas/index', $data);
    }

    public function tambah()
    {
        $this->form_validation->set_rules( 'kode', 'Kode', 'required|is_unique[tb_pengawas.kode_pengawas]' );
        $this->form_validation->set_rules( 'nama', 'Nama', 'required' );

        $data['title'] = 'Tambah pengawas';

        if ($this->form_validation->run() === FALSE)
        {
            load_view('pengawas/tambah', $data);
        }
    }
}

```



```

    }
    else
    {
        $fields = array(
            'kode_pengawas' => $this->input->post('kode'),
            'nama_pengawas' => $this->input->post('nama'),
            'keterangan' => $this->input->post('keterangan'),
        );
        $this->pengawas_model->tambah($fields);
        redirect('pengawas');
    }
}

public function ubah( $ID = null )
{
    $this->form_validation->set_rules( 'nama', 'Nama', 'required' );

    $data['title'] = 'Ubah pengawas';

    if ( $this->form_validation->run() === FALSE )
    {
        $data['row'] = $this->pengawas_model->get_pengawas($ID);
        load_view('pengawas/ubah', $data);
    }
    else
    {
        $fields = array(
            'nama_pengawas' => $this->input->post('nama'),
            'keterangan' => $this->input->post('keterangan'),
        );
        $this->pengawas_model->ubah($fields, $ID);
        redirect('pengawas');
    }
}

public function hapus( $ID = null )
{
    $this->pengawas_model->hapus($ID);
    redirect('pengawas');
}
}

```

Gambar 4. 31 *Script* Data Pengawas

Gambar 4.31. merupakan script untuk menampilkan data pengawas yang sudah pernah di inputkan. Serta dapat menambah data pengawas dan mengubah data pengawas yang sudah ada.

g. *Script* Proses Penjadwalan

```

<?php
class AG {

    var $generation = 0; //generasi ke...
    var $success = false; //keadaan jika sudah ada solusi terbaik
    var $console = "";
    var $best_cromossom = array();
    var $best_fitness = 0;

    function __construct($matkul, $waktu, $pengawas, $ruang) {

```

```

foreach
    $this->matkul[] = array(
        'kode_matkul' => $row->kode_matkul,
        'peserta' => $a < $max_ruang ? $a : $max_ruang,
        'waktu' => $this->>waktu[$index_waktu]->id_waktu,
    );
    $a=$max_ruang;
}
} else {
    $this->matkul[] = array(
        'kode_matkul' => $row->kode_matkul,
        'peserta' => $row->peserta,
        'waktu' => $this->>waktu[$index_wa($ruang as $row){
    $this->ruang[$row->kode_ruang] = $row->kapasitas;
}
}

$max_ruang = max($this->ruang);
$min_ruang = min($this->ruang);

foreach($waktu as $row){
    $this->>waktu[] = $row;
}

$index_waktu = 0;
foreach($matkul as $row){
    if($index_waktu==count($this->>waktu))
        $index_waktu = 0;

    if($row->peserta > $max_ruang){
        $a = $row->peserta;
        while($a > 0){
            $r = $row;ktu]->id_waktu,
        );
    }
    $index_waktu++;
}
foreach($pengawas as $row){
    $this->pengawas[$row->kode_pengawas] = $row->nama_pengawas;
}
//echo '<pre>' . print_r($this->matkul, 1) . '</pre>';
}

function generate(){

    $this->time_start = microtime(true);

    $this->generate_crommosom();

    while(($this->generation < $this->max_generation) && $this->success == false){
        $this->generation++;
        $this->console.= "<h4>Generasi ke-$this->generation</h4>";
        $this->show_crommosom();
        $this->calculate_all_fitness();
        $this->show_fitness();

        // 1 -----
        if(!$this->success) {
            $this->crossover();
            $this->show_crommosom();
            $this->show_fitness();
        }

        // 2 -----
        if(!$this->success) {

```

```

    $this->mutation();
    $this->show_crommosom();
    $this->show_fitness();
}

/// 3 -----
if(!$this->success) {
    $this->get_com_pro(); //evaluasi

    $this->selection();
    $this->show_crommosom();
    $this->show_fitness();
}

}
$this->save_result();
$this->time_end = microtime(true);
$time = $this->time_end - $this->time_start;

echo "<pre style='font-size:0.8em'>\nFITNESS TERBAIK: $this->best_fitness";
// echo "\nExecution Time: $time seconds";
//echo "\nMemory Usage: " . memory_get_usage() / 1024 . ' kilo bytes';
echo "\nGENERASI: $this->generation";
//echo "\nCROMOSSOM TERBAIK: " . $this->print_cros($this->best_cromossom) . "</pre>";
$this->get_debug();
}

/// crossover -----
/*Crossover merupakan operasi yang bekerja untuk menggabungkan 2 kromosom induk(parent)
menjadi kromosom baru. Jumlah solusi jadwal yang mengalami crossover
ditentukan oleh parameter crossover rate. Metode crossover yang paling umum
digunakan adalah metode one-cut-point, yang secara acak memilih satu titik potong
dan menukarkan bagian kanan dari tiap induk untuk menghasilkan offspring (Mahmudy, 2015).*/
function crossover(){
    $this->console.= "<h5>Pindah silang generasi ke-$this->generation</h5>";
    $parent = array();

    foreach($this->crommosom as $key => $val) {
        $rnd = mt_rand() / mt_getrandmax();
        if($rnd <= $this->crossover_rate / 100)
            $parent[] = $key;
    }
    //reset($this->crommosom);

    foreach($parent as $key => $val) {
        $this->console.="Parent[$key] : $val \n";
    }

    $parent = $parent;
    $c = count($parent);

    if( $c > 1 ) {
        for($a = 0; $a < $c-1; $a++) {
            $new_cro[$parent[$a]] = $this->get_crossover( $parent[$a], $parent[$a+1]);
        }
        $new_cro[$parent[$c-1]] = $this->get_crossover( $parent[$c-1], $parent[0]);

        foreach($new_cro as $key => $val) {
            $this->crommosom[$key] = $val;
            $this->calculate_fitness($key);
        }
    }
}
}

```

```

function get_crossover($key1, $key2){
    $cro1 = $this->crommosom[$key1];
    $cro2 = $this->crommosom[$key2];

    $offspring = rand(0, count($cro1) - 2);
    $new_cro = array();

    for($a = 0; $a < count($cro1); $a++) {
        if( $a <= $offspring )
            $new_cro[] = $cro1[$a];
        else
            $new_cro[] = $cro2[$a];
    }

    //$this->console.= "Offspring: $offspring \n";

    return $new_cro;
}

/// mutation -----
function mutation(){
    $mutation = array();
    $this->console.= "<h5>Mutasi generasi ke-$this->generation</h5>";
    $gen_per_cro = count($this->matkul);
    $total_gen = count($this->crommosom) * $gen_per_cro;
    $total_mutation = ceil($this->mutation_rate / 100 * $total_gen);

    for($a = 1; $a <= $total_mutation; $a++) {
        $val = rand(1, $total_gen);

        $cro_index = ceil($val / $gen_per_cro) - 1;
        $gen_index = (( $val - 1) % $gen_per_cro);

        $this->console.= "$val : [$cro_index, $gen_index]\n";
        $this->crommosom[$cro_index][$gen_index]['waktu'] = array_rand($this->waktu);
        $this->crommosom[$cro_index][$gen_index]['pengawas'] = array_rand($this->pengawas);
        $this->crommosom[$cro_index][$gen_index]['ruang'] = array_rand($this->ruang);
        $this->fitness[$cro_index] = $this->calculate_fitness($cro_index);

        if($this->success)
            return;
    }
    return false;
}

/// selection -----
function selection(){
    $this->console.= "<h5>Seleksi generasi ke-$this->generation</h5>";
    $this->get_rand();
    $new_cro = array();
    foreach ($this->rand as $key => $val) {
        $k = $this->choose_selection($val);
        $new_cro[$key] = $this->crommosom[$k];
        $this->fitness[$key] = $this->fitness[$val];
        //$this->console.= "K[$key] = K[$k] \n";
    }
    $this->crommosom = $new_cro;
}

function choose_selection($rand_numb = 0) {
    foreach($this->com_pro as $key => $val) {
        if($rand_numb <= $val)
            return $key;
    }
}

```

```

/// show_crommosom -----
function show_crommosom() {
    $a = array();
    foreach ($this->crommosom as $key => $val) {
        $a[] = $this->print_cros($val, $key);
    }
    //echo '<pre>' . print_r($this->crommosom, 1) . '</pre>';
    $this->console.= implode("\n", $a) . "\n";
}

function print_cros($val = array(), $key = 0){
    $arr = array();
    foreach($val as $k => $v) {
        $arr[] = '[' . implode(', ', $v) . ']';
    }
    return "Kromosom[$key]: (" . implode(', ', $arr) . ")";
}

/// calculate_fitness -----
function calculate_fitness($key) {
    $cro = $this->crommosom[$key];
    //pengawas sama waktu sama
    $clash_pengawas = $this->get_clash_pengawas($cro);
    //ruang sama waktu sama
    $clash_ruang = $this->get_clash_ruang($cro);
    //jumlah melebihi kapasitas
    $clash_jumlah = $this->get_clash_jumlah($cro);

    $f['desc'] = "1/(1+$clash_pengawas+$clash_ruang+$clash_jumlah)";
    $f['nilai'] = 1/ (1 + $clash_pengawas + $clash_ruang + $clash_jumlah);

    if($f['nilai']==1) // jika sudah optimal maka berhenti
    $this->success = true;

    if($f['nilai'] > $this->best_fitness) {
        $this->best_fitness = $f['nilai'];
        $this->best_crommosom = $this->crommosom[$key];
    }

    $this->fitness[$key] = $f;
    return $f;
}

function calculate_all_fitness() {
    foreach($this->crommosom as $key => $val) {
        $this->calculate_fitness($key);
    }
}

/// crommosom -----
function generate_crommosom() {
    $numb = 0;
    while($numb < $this->num_crommosom) {
        $cro = $this->get_rand_crommosom();
        $this->crommosom[] = $cro;
        $this->fitness[] = 0;
        $numb++;
    }
    //echo '<pre>' . print_r($this->crommosom, 1) . '</pre>';
    //print_r($this->fitness);
}

function get_rand_crommosom(){
    $arr = array();

```

```

foreach($this->matkul as $key => $value) {
    $arr[] = array(
        'matkul' => $key,
        'pengawas' => array_rand($this->pengawas),
        'ruang' => array_rand($this->ruang),
    );
}
//echo '<pre>'. print_r($arr, 1). '</pre>';
return $arr;
}

/// show_fitness -----
function get_total_fitness(){
    $this->total_fitness = 0;
    //reset($this->fitness);
    foreach($this->fitness as $val) {
        $this->total_fitness+=$val['nilai'];
    }
    return $this->total_fitness;
}
function show_fitness(){
    foreach($this->fitness as $key => $fit) {
        $this->console.= "F[$key]: $fit[desc] = $fit[nilai] \n";
    }
    //reset($this->fitness);
    $this->get_total_fitness();
    $this->console.= "Total F: ".$this->total_fitness . "\n";
}

/// get_probability -----
/// EVALUASI
function get_probability(){
    $this->probability = array();
    foreach($this->fitness as $key => $val) {
        $x = $val['nilai'] / $this->total_fitness;
        $this->probability[] = $x;
        //$this->console.= "P[$key] : $x \n";
    }
    //$this->console.= "Total P: " . array_sum($this->probability) . "\n";
    //reset($this->fitness);
    return $this->probability;
}

function get_com_pro(){ //evaluasi

    $this->get_probability();

    $this->com_pro = array();
    $x = 0;
    foreach($this->probability as $key => $val) {
        $x+= $val;
        $this->com_pro[] = $x;
        //$this->console.= "PK[$key] : $x \n";
    }
    //reset($this->probability);
    $this->com_pro;
}

function get_rand(){
    $this->rand = array();
    //reset($this->fitness);
    foreach($this->fitness as $key => $val) {
        $r = mt_rand() / mt_getrandmax();
        $this->rand[] = $r;
    }
}

```

```

    // $this->console.="R[$key] : $r \n";
  }
}

/// 3 RULE -----
function get_clash_ruang($crom = array()) {
  $clash = 0;
  $count = count($crom);
  for($a = 0; $a < $count - 1; $a++) {
    for($b = $a + 1; $b < $count; $b++) {
      if($crom[$a]['ruang']==$crom[$b]['ruang']) {
        if($this->matkul[$a]['waktu'] == $this->matkul[$b]['waktu']){
          $clash++;
        }
      }
    }
  }
  return $clash;
}

function get_clash_jumlah($crom = array()) {
  $clash = 0;
  foreach($crom as $key => $val){
    $kapasitas = $this->ruang[$val['ruang']];
    $peserta = $this->matkul[$val['matkul']]['peserta'];

    //echo $peserta.'>'. $kapasitas.'<br />';
    if($peserta > $kapasitas)
      $clash++;
  }
  return $clash;
}

function get_clash_pengawas($crom = array()) {
  $clash = 0;
  $count = count($crom);
  for($a = 0; $a < $count - 1; $a++) {
    for($b = $a + 1; $b < $count; $b++) {
      if($crom[$a]['pengawas']==$crom[$b]['pengawas']) {
        if($this->matkul[$a]['waktu'] == $this->matkul[$b]['waktu']){
          $clash++;
        }
      }
    }
  }
  return $clash;
}

/// 3 RULE -----
function get_debug(){
  if($this->debug)
    echo "<pre style='font-size:0.8em'>$this->console</pre>";
}

function save_result(){
  $CI =& get_instance();

  // $CI->db->query("DELETE FROM tb_jadwal");
  $CI->db->query("TRUNCATE TABLE tb_jadwal");

  //echo '<pre>' . print_r($this->best_cromossom, 1) . '</pre>';

  foreach($this->best_cromossom as $key => $val){
    $matkul = $this->matkul[$val['matkul']]['kode_matkul'];
  }
}

```

```

    $peserta = $this->matkul[$val['matkul']]['peserta'];
    $waktu = $this->matkul[$val['matkul']]['waktu'];

    $CI->db->query("INSERT INTO tb_jadwal (matkul, peserta, waktu, pengawas, ruang)
    VALUES ( '$matkul', '$peserta', '$waktu', '$val[pengawas]', '$val[ruang]'");
  }
  reset($this->best_cromossom);
}
}
}

```

Gambar 4. 32 *Script* Data Proses Penjadwalan

Gambar 4.32. merupakan script untuk proses penjadwalan.

h. *Script* Hasil Jadwal

```

<?php
class Jadwal extends CI_Controller{

    public function __construct()
    {
        parent::__construct();
        $this->load->model('waktu_model');
        $this->load->model('matkul_model');
        $this->load->model('pengawas_model');
        $this->load->model('ruang_model');
        $this->load->model('jadwal_model');
    }

    public function index()
    {
        $data['title'] = 'Jadwal Ujian';
        $data['q'] = $this->input->get('search');
        $data['rows'] = $this->jadwal_model->tampil('jadwal', $data['q']);
        load_view('jadwal/index', $data);
    }

    public function ubah( $ID = null )
    {
        $this->form_validation->set_rules( 'kuliah', 'Ujian', 'required' );
        $this->form_validation->set_rules( 'ruang', 'Ruang', 'required' );
        $this->form_validation->set_rules( 'waktu', 'Waktu', 'required' );

        $data['title'] = 'Ubah jadwal';

        if ( $this->form_validation->run() === FALSE )
        {
            $data['row'] = $this->jadwal_model->get_jadwal($ID);
            load_view('jadwal/ubah', $data);
        }
        else
        {
            $fields = array(
                'kuliah' => $this->input->post('kuliah'),
                'ruang' => $this->input->post('ruang'),
                'waktu' => $this->input->post('waktu'),
            );
            $this->jadwal_model->ubah($fields, $ID);
            redirect('jadwal/hasil');
        }
    }

    public function acc(){
        $this->db->query("DELETE FROM tb_jadwal WHERE status='jadwal'");
        $this->db->query("UPDATE tb_jadwal SET status='jadwal'");
        redirect('jadwal');
    }
}

```



```

}

public function hasil()
{
    $data['title'] = 'Hasil Penjadwalan';
    $data['rows'] = $this->jadwal_model->tampil('hasil', $this->input->get('search'));
    load_view('jadwal/hasil', $data);
}

public function cetak( $ID = NULL)
{
    $data['rows'] = $this->jadwal_model->tampil('jadwal', $this->input->get('search'));
    load_view_cetak('jadwal/cetak', $data);
}

public function hitung()
{
    $data['title'] = 'Generate Jadwal';

    $data['arrMatkul'] = $this->matkul_model->tampil();
    $data['arrWaktu'] = $this->>waktu_model->tampil();
    $data['arrPengawas'] = $this->pengawas_model->tampil();
    $data['arrRuang'] = $this->ruang_model->tampil();

    $this->form_validation->set_rules( 'kode', 'Kode', 'required|is_unique[tb_dosen.kode_dosen]' );
    $this->form_validation->set_rules( 'nama', 'Nama', 'required' );

    if ($this->form_validation->run() === FALSE)
    {
        load_view('jadwal/form', $data);
    }
    else
    {
        load_view('jadwal/form', $data);
    }
}
}

```

Gambar 4. 33 *Script* Hasil Jadwal

Gambar 4.33. merupakan script untuk menampilkan data hasil jadwal yang sudah pernah di inputkan. Serta dapat mencetak jadwal yang sudah ada.

B. PENGUJIAN

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian adalah menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal yaitu mampu mempretasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengekodean dari perangkat lunak itu sendiri.

1. Pengujian Matakuliah

Pengujian ini menguji sistem penjadwalan otomatis dapat menjadwalkan satu matakuliah diujikan pada waktu yang sama tetapi berbeda ruang. Pengujian ini dapat dilihat dari tabel 4.3

Tabel 4. 3 Tabel Pengujian Matakuliah

No	Hari, tanggal ujian	Kode Matakuliah	Nama Matakuliah	Jam	Ruang ujian	Peserta
1	Senin, 27 januari 2020	KPT0601202	Psikologi D3/1	07.00-08.30	Ruangan 309	30
2	Senin, 27 januari 2020	KPT0601202	Psikologi D3/1	07.00-08.30	Ruangan 308	17
3	Senin, 27 januari 2020	KPT060301	Falsafah dan Teori Keperawatan S1/1	08.45-10.15	Ruangan 309	30

Sumber : data diolah 2020

Kapasitas ruang ujian yang ada di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang hanya dapat menampung 30 peserta ujian, sedangkan peserta ujian untuk setiap matakuliah rata-rata lebih dari 30 peserta. Matakuliah yang peserta ujian lebih dari kapasitas ruang ujian maka membutuhkan lebih dari 1 ruang ujian. Matakuliah Psikologi dengan kode matakuliah KPT0601202 terdapat 47 peserta ujian, sehingga ruang ujian untuk matakuliah psikologi membutuhkan 2 ruang ujian. Tabel 4.3 matakuliah psikologi diujikan di ruang 309 dan ruang 309 serta dijadwalkan untuk diujikan pada hari yang sama yaitu senin 27 januari 2020 jam 07.00 – 08.30. Pembagiannya ruang 309 sebanyak 30 peserta ujian dan 309 sebanyak 17 peserta ujian.

Matakuliah Falsafah dan Teori Keperawatan dengan kode matakuliah KPT060301 terdapat 30 peserta ujian, sehingga ruang ujian untuk matakuliah Falsafah dan Teori Keperawatan membutuhkan 1 ruang ujian. Tabel 4.3 matakuliah Falsafah dan Teori Keperawatan diujikan di

ruang 309 serta dijadwalan untuk diujikan pada hari senin 27 januari 2020 jam 08.45 – 10.15. Hasil pengujian tersebut dapat disimpulkan bahwa sistem penjadwalan ini dapat melakukan pembagian peserta ujian dan penentuan ruang ujian secara otomatis serta sistem dapat menjadwalkan satu matakuliah dilakukan ujian dalam satu waktu yang bersamaan.

2. Pengujian Pengawas

Pengujian ini menguji sistem penjadwalan otomatis dapat menjamin seorang pengawas tidak mengawasi ujian dalam waktu yang bersamaan. Pengujian ini dapat dilihat dari tabel 4.4

Tabel 4. 4 Tabel Pengujian Matakuliah

No	Hari, tanggal ujian	Kode Matakuliah	Nama Matakuliah	Jam	Ruang ujian	Pengawas	Peserta
1	Senin, 27 januari 2020	KPT0601202	Psikologi	07.00-08.30	Ruangan 309	Nurul Niawati	30
2	Senin, 27 januari 2020	KPT0601202	Psikologi	07.00-08.30	Ruangan 308	Prasetyo Budi	17

Sumber : data diolah 2020

Jumlah pengawas ujian yang ada di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang hanya 5 orang, sedangkan jumlah ruangan yang harus diawasi kurang lebih 10 ruangan setiap harinya. Dihawatirkan akan terjadi tabrakan pengawas dikarenakan jumlah pengawas yang terlalu sedikit, karena pengawas hanya diperbolehkan mengawasi 1 matakuliah dalam satu waktu dan satu ruangan ujian. Pengawas dengan nama Nurul Niawati hanya dapat mengawasi 1 matakuliah dalam 1 waktu dan 1 ruangan ujian. Tabel 4.4 Pengawas Nurul Niawati mengawasi matakuliah Psikologi dengan kode matakuliah KPT0601202 dijadwalan untuk diujikan pada hari senin 27 januari 2020 jam 07.00 – 08.30 pada ruangan 309.

Pengawas dengan nama Prasetyo Budi hanya dapat mengawasi 1 matakuliah dalam 1 waktu dan 1 ruangan ujian. Tabel 4.4 Pengawas

Prasetyo Budi mengawasi matakuliah Psikologi dengan kode matakuliah KPT0601202 dijadwalan untuk diujikan pada hari senin 27 januari 2020 jam 07.00 – 08.30 pada ruangan 308. Hasil pengujian tersebut dapat disimpulkan bahwa sistem penjadwalan otomatis ini dapat melakukan pembagian pengawas hanya mengawasi 1 ruangan dalam 1 waktu ujian.

3. Pengujian Penjadwalan

Pengujian ini menguji sistem penjadwalan otomatis jadwal ujian dua prodi dibuat secara bersamaan. Pengujian ini dapat dilihat dari tabel 4.5

Tabel 4. 5 Tabel Pengujian penjadwalan

No	Hari, tanggal ujian	Kode Matakuliah	Nama Matakuliah	Jam	Ruang ujian	Pengawas	Peserta
1	Senin, 27 januari 2020	KPT0601202	Psikologi D3/1	07.00-08.30	Ruangan 309	Nurul Niawati	30
2	Senin, 27 januari 2020	KPT0601202	Psikologi D3/1	07.00-08.30	Ruangan 308	Prasetyo Budi	17
3	Senin, 27 januari 2020	KPT0603301	Falsafah dan Teori Keperawatan S1/1	08.45-10.15	Ruangan 309	Guno Ariyanto	30

Sumber : data diolah 2020

Waktu pelaksanaan ujian yang ada di Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang hanya 12 hari sedangkan ada sekitar 43 matakuliah yang diujikan, matakuliah itu diambil dari 2 prodi yaitu prodi keperawatan D3 dan prodi keperawatan S1, Pembuatan jadwal ujian masing-masing prodi dibuat sendiri-sendiri dengan jumlah waktu sama, jumlah pengawas sama dan jumlah ruangan yang sama. Hal ini menyebabkan sering terjadinya bentrokan waktu, pengawas dan ruang ujian. Sistem akan membuat 1 jadwal ujian yang didalamnya terdiri dari sekumpulan Matakuliah dari 2 prodi untuk menjamin jadwal tidak bertabrakan.

Penjadwalan 2 prodi secara bersamaan dapat dilakukan dengan cara memasukkan data matakuliah dari dua prodi kedalam sistem penjadwalan ujian otomatis, dengan melakukan hal tersebut sistem akan

membagi hari, waktu, pengawas, dan ruangan masing-masing untuk 1 jadwal ujian akhir semester yang didalamnya terdapat sejumlah matakuliah yang diujikan dari 2 prodi.

Jadwal yang dihasilkan setelah proses penjadwalan dengan matakuliah psikologi yang merupakan matakuliah dari prodi keperawatan D3 dijadwalkan dalam 1 hari yang sama, pengawas berbeda, jam sama dan ruangan berbeda. Tabel 4.5 matakuliah psikologi dengan kode matakuliah KPT0601202 diujikan di ruang 306 dan ruang 308 serta dijadwalkan untuk diujikan pada hari yang sama yaitu senin 27 januari 2020 pada jam 07.00 – 08.30 dengan pengawas ruangan 309 Nurul Niawati dan pengawas ruangan 308 Prasetyo Budi.

Matakuliah konsep dasar keperawatan yang merupakan jadwal keperawatan S1 dijadwalkan dalam 1 hari yang sama, pengawas berbeda, jam sama dan ruangan berbeda. Tabel 4.5 matakuliah Falsafah dan Teori Keperawatan dengan kode matakuliah KPT0603301 diujikan di ruang 307 serta dijadwalkan untuk diujikan pada hari yang sama yaitu senin 27 januari 2020 pada jam 08.45 – 10.15 dengan pengawas ruangan 309 Guno Ariyanto. Hasil pengujian tersebut dapat disimpulkan bahwa sistem penjadwalan otomatis ini dapat melakukan pembagian jadwal ujian yang terdiri dari 2 prodi secara bersamaan.

4. Pengujian Black Box

Tabel 4. 6 Pengujian Black Box

Item Uji	Detail Pengujian	Pertanyaan	Jawaban	
			Ya	Tidak
Menu Login				
Login	Masukkan username	Dapatkah memasukkan username	✓	
	Masukkan password	Dapatkah memasukkan password	✓	
	Menekan tombol masuk	Apakah tombol masuk berfungsi dengan benar	✓	
		Adakah Keterangan Apabila username dan password salah saat menekan tombol masuk	✓	
Menu User				
Tambah User	Menambah Username	Dapatkah menambah username baru	✓	
	Menambah Password	Dapatkah menambah password baru	✓	
	Memilih Level	Dapatkah memilih level	✓	
	Menekan Tombol Simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah User	Merubah username	Dapatkah merubah username lama	✓	
	Merubah Password	Dapatkah merubah password lama	✓	
	Memilih Level	Dapatkah memilih level	✓	
	Menekan Tombol Simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Hapus User	Menekan tombol Hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu MataKuliah				
Tambah	Masukkan Kode	Dapatkah Menambah Kode	✓	

Makul		Baru		
	Masukkan nama matakuliah	Dapatkah menambah nama matakuliah baru	✓	
	Masukkan SKS	Dapatkah menambah sks baru	✓	
	Masukkan Peserta	Dapatkah menambah peserta baru	✓	
	Menekan Tombol Simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah Makul	Merubah kode	Dapatkah merubah kode baru	✓	
	Merubah nama matakuliah	Dapatkah merubah nama matakuliah baru	✓	
	Merubah sks	Dapatkah merubah sks baru	✓	
	Merubah peserta	Dapatkah merubah peserta baru	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Hapus Makul	Menekan tombol hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu Pengawas				
Tambah Pengawas	Masukkan kode	Dapatkah menambah kode baru	✓	
	Masukkan nama pengawas	Dapatkah menambah nama pengawas baru	✓	
	Masukkan keterangan	Dapatkah menambah keterangan	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah Pengawas	Merubah kode	Dapatkah merubah kode	✓	
	merubah nama pengawas	Dapatkah merubah nama pengawas	✓	
	merubah keterangan	Dapatkah merubah keterangan	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	

Hapus Makul	Menekan tombol hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu Waktu				
Tambah Waktu	Masukkan tanggal	Dapatkah memilih tanggal	✓	
	Masukkan jam	Dapatkah memilih jam	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah Waktu	Merubah tanggal	Dapatkah memilih tanggal	✓	
	Merubah jam	Dapatkah memilih jam	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Hapus Waktu	Menekan tombol hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu Jam				
Tambah Jam	Masukkan kode	Dapatkah menambah kode baru	✓	
	Masukkan jam mulai	Dapatkah menambah jam mulai baru	✓	
	Masukkan jam selesai	Dapatkah menambah jam selesai baru	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah Jam	Merubah kode	Dapatkah merubah kode lama	✓	
	Merubah jam mulai	Dapatkah merubah jam mulai lama	✓	
	Merubah jam selesai	Dapatkah merubah jam selesai lama	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Hapus Jam	Menekan tombol hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu Ruang				

Tambah Ruang	Masukkan kode	Dapatkah Menambah kode baru	✓	
	Masukkan nama ruang	Dapatkah menambah nama ruangan baru	✓	
	Masukkan kapasitas	Dapatkah menambah kapasitas	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Ubah Ruang	Merubah kode	Dapatkah merubah kode lama	✓	
	Merubah nama ruang	Dapatkah merubah nama ruang lama	✓	
	Merubah kapasitas	Dapatkah merubah kapasitas lama	✓	
	Menekan tombol simpan	Apakah tombol simpan berfungsi	✓	
	Menekan tombol kembali	Apakah tombol kembali berfungsi	✓	
Hapus Ruang	Menekan tombol hapus	Apakah tombol hapus berfungsi	✓	
		Apakah data terhapus	✓	
Menu Penjadwalan				
Generate	Menekan tombol generate	Apakah tombol generate berfungsi	✓	
Menu Jadwal				
Cetak	Menekan tombol cetak	Apakah tomol cetak berfungsi	✓	
Menu Loguot				
	Menekan tombol logout	Apakah tombol logout berfungsi	✓	

Keterangan :

Ya : Valid

Tidak : Tidak Valid

5. Acceptance Testing

Pada tahap ini dilakukan pengujian yang sifatnya menyeluruh dari semua data yang ada dengan melakukan penilaian secara subyektif dari semua kalangan yang menggunakan sistem. Pengujian ini dilakukan testing menyeluruh terhadap pengguna sistem yang meliputi antara lain

3 staff TU Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang. Adapun pokok pertanyaan dari rancangan Acceptance Testing sebagai berikut.

- a. Apakah fungsi dari sistem dapat berjalan secara keseluruhan?
- b. Apakah sistem yang telah dibuat bermanfaat bagi Fakultas IlmuKesehatan Muhammadiyah Magelang dalam pembuatan jadwal ujian akhir semester?
- c. Apakah pengguna sistem merasakan manfaat dari adanya sistem penjadwalan ujian yang dibuat, memudahkan TU dalam pembuatan jadwal ujian akhir semester ?

BAB V

HASIL DAN PEMBAHASAN

A. HASIL

1. Hasil Implementasi Sistem

Pengujian satu dilakukan menggunakan aplikasi dengan memasukkan data penjadwalan ujian semester ganjil fakultas ilmu kesehatan tahun ajaran 2019/2020. proses input dapat dilihat pada gambar 5.1. Data pengujian menggunakan aplikasi adalah sebagai berikut :

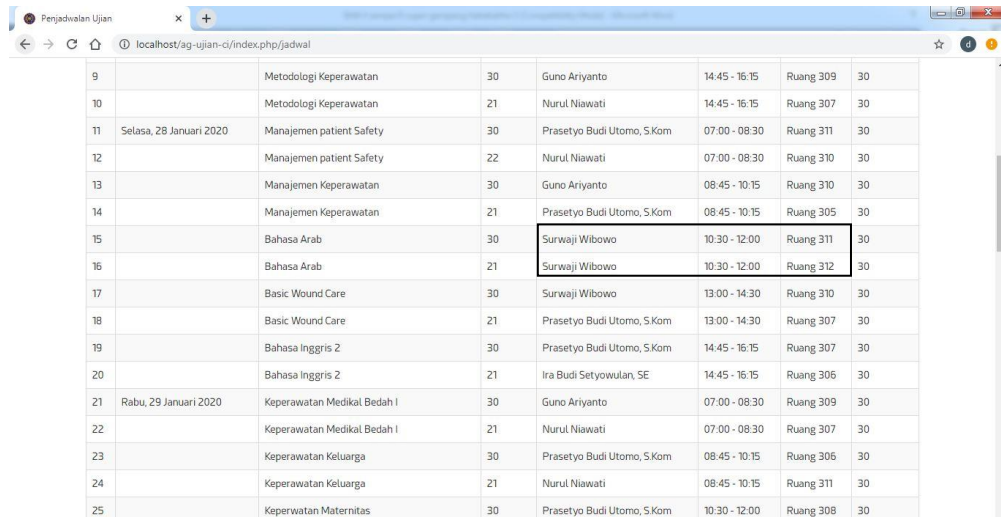
The screenshot shows a web browser window with the URL `localhost/ag-ujian-ci/index.php/jadwal/hitung`. The page header includes the logo of Universitas Muhammadiyah Magelang and the title 'Algoritma Genetika Penjadwalan Ujian'. The main content area is titled 'Generate Jadwal' and contains the following elements:

- Jumlah Kromosom Dibangkitkan:** Input field containing '10', with a note 'Masukkan antara 10-500'.
- Maksimal Generasi:** Input field containing '25', with a note 'Masukkan antara 25-500'.
- Buttons:** 'Generate Jadwal' (blue) and 'Lihat Jadwal' (green).
- Output:** A box displaying 'FITNESS TERBAIK: 0,25' and 'GENERASI: 25'.

The footer of the page contains the copyright information: '© 2020 Algoritma Genetika Penjadwalan Ujian, Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang'.

Gambar 5. 1 Hasil Jadwal Uji ke satu dengan Aplikasi Algoritma Genetika

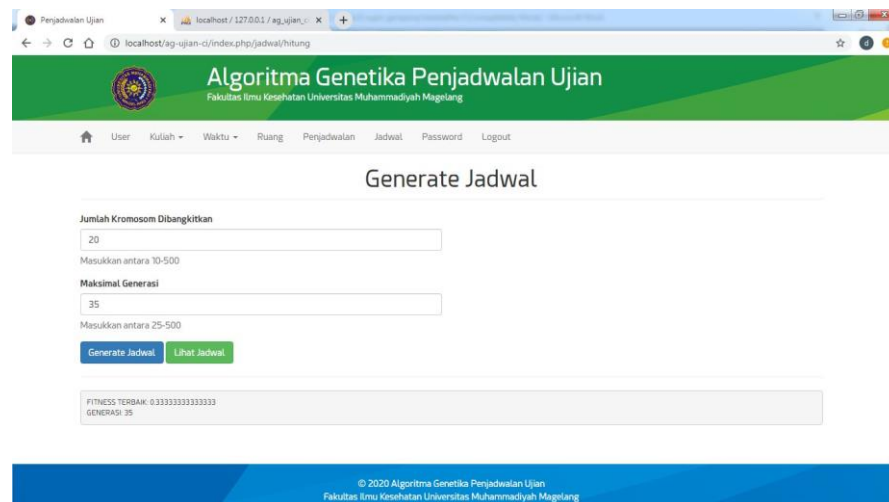
Hasil pengujian dilakukan dengan Jumlah data matakuliah 45, jumlah ruangan 8, jumlah hari 10, dan 5 pengawas dengan jumlah kromosom 10 dan banyak generasi 25 hasilnya belum optimal dengan nilai fitness 0,25. Hasil dari penjadwalan dapat dilihat pada gambar 5.2.



9		Metodologi Keperawatan	30	Guno Ariyanto	14:45 - 16:15	Ruang 309	30
10		Metodologi Keperawatan	21	Nurul Niawati	14:45 - 16:15	Ruang 307	30
11	Selasa, 28 Januari 2020	Manajemen patient Safety	30	Prasetyo Budi Utomo, S.Kom	07:00 - 08:30	Ruang 311	30
12		Manajemen patient Safety	22	Nurul Niawati	07:00 - 08:30	Ruang 310	30
13		Manajemen Keperawatan	30	Guno Ariyanto	08:45 - 10:15	Ruang 310	30
14		Manajemen Keperawatan	21	Prasetyo Budi Utomo, S.Kom	08:45 - 10:15	Ruang 305	30
15		Bahasa Arab	30	Surwaji Wibowo	10:30 - 12:00	Ruang 311	30
16		Bahasa Arab	21	Surwaji Wibowo	10:30 - 12:00	Ruang 312	30
17		Basic Wound Care	30	Surwaji Wibowo	13:00 - 14:30	Ruang 310	30
18		Basic Wound Care	21	Prasetyo Budi Utomo, S.Kom	13:00 - 14:30	Ruang 307	30
19		Bahasa Inggris 2	30	Prasetyo Budi Utomo, S.Kom	14:45 - 16:15	Ruang 307	30
20		Bahasa Inggris 2	21	Ira Budi Setyowulan, SE	14:45 - 16:15	Ruang 306	30
21	Rabu, 29 Januari 2020	Keperawatan Medikal Bedah I	30	Guno Ariyanto	07:00 - 08:30	Ruang 309	30
22		Keperawatan Medikal Bedah I	21	Nurul Niawati	07:00 - 08:30	Ruang 307	30
23		Keperawatan Keluarga	30	Prasetyo Budi Utomo, S.Kom	08:45 - 10:15	Ruang 306	30
24		Keperawatan Keluarga	21	Nurul Niawati	08:45 - 10:15	Ruang 311	30
25		Keperawatan Maternitas	30	Prasetyo Budi Utomo, S.Kom	10:30 - 12:00	Ruang 308	30

Gambar 5. 2 Hasil penjadwalan dengan jumlah kromosom 10 dan maksimal generasin 25.

Pengujian dua dilakukan menggunakan aplikasi dengan memasukkan data penjadwalan ujian akhir semester ganjil fakultas ilmu kesehatan ajaran 2019/2020. proses input dapat dilihat pada gamabar 5.3. Data pengujian menggunakan aplikasi adalah sebagai berikut :



Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

User Kuliah Waktu Ruang Penjadwalan Jadwal Password Logout

Generate Jadwal

Jumlah Kromosom Dibangkitkan
20
Masukkan antara 10-500

Maksimal Generasi
35
Masukkan antara 25-500

Generate Jadwal Lihat Jadwal

FITNESS TERBAIK: 0,3333333333333333
GENERASI: 35

© 2020 Algoritma Genetika Penjadwalan Ujian
Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang

Gambar 5. 3 Hasil Jadwal Uji ke dua dengan Aplikasi Algoritma Genetika.

Hasil pengujian dilakukan dengan Jumlah data matakuliah 45, jumlah ruangan 8, jumlah hari 10, dan 5 pengawas dengan jumlah kromosom 20 dan banyak generasi 35 hasilnya belum optimal dengan nilai fitness 0,3333. Hasil dari penjadwalan dapat dilihat pada gamabar 5.4.

No	Hari, Tanggal	Mata Kuliah	Peserta	Pengawas	Jam	Ruang	Kapasitas
1	Senin, 27 Januari 2020	Psikologi	30	Ira Budi Setyowulan, SE	07.00 - 08.30	Ruang 307	30
2		Psikologi	17	Guno Ariyanto	07.00 - 08.30	Ruang 311	30
3		Konsep Dasar Keperawatan	30	Guno Ariyanto	08.45 - 10.15	Ruang 312	30
4		Konsep Dasar Keperawatan	17	Nurul Niawati	08.45 - 10.15	Ruang 308	30
5		Gizi Dan Diet	30	Surwaji Wibowo	10.30 - 12.00	Ruang 305	30
6		Gizi Dan Diet	17	Nurul Niawati	10.30 - 12.00	Ruang 310	30
7		Antropologi Kesehatan	30	Guno Ariyanto	13.00 - 14.30	Ruang 307	30
8		Antropologi Kesehatan	18	Surwaji Wibowo	13.00 - 14.30	Ruang 311	30
9		Metodologi Keperawatan	30	Surwaji Wibowo	14.45 - 16.15	Ruang 307	30
10		Metodologi Keperawatan	21	Guno Ariyanto	14.45 - 16.15	Ruang 308	30
11	Selasa, 28 Januari 2020	Manajemen patient Safety	30	Guno Ariyanto	07.00 - 08.30	Ruang 310	30
12		Manajemen patient Safety	22	Nurul Niawati	07.00 - 08.30	Ruang 305	30
13		Manajemen Keperawatan	30	Nurul Niawati	08.45 - 10.15	Ruang 309	30
14		Manajemen Keperawatan	21	Nurul Niawati	08.45 - 10.15	Ruang 308	30

Gambar 5. 4 Hasil penjadwalan dengan jumlah kromosom 20 dan maksimal generasin 35.

Pengujian tiga dilakukan menggunakan aplikasi dengan memasukkan data penjadwalan ujian semester ganjil fakultas ilmu kesehatan tahun ajaran 2019/2020. proses input dapat dilihat pada gamabar 5.5. Data pengujian menggunakan aplikasi adalah sebagai berikut :

Gambar 5. 5 Hasil Jadwal Uji ke tiga dengan Aplikasi Algoritma Genetika

Hasil pengujian dilakukan dengan Jumlah data matakuliah 45, jumlah ruangan 8, jumlah hari 10, dan 5 pengawas dengan jumlah kromosom 30 dan banyak generasi 35 hasilnya belum optimal dengan nilai fitness 0,5. Hasil dari penjadwalan dapat dilihat pada gambar 5.6.

No	Matakuliah	Kromosom	Pengawas	Waktu	Ruangan	Kapasitas
17	Basic Wound Care	21	Ira Budi Setyowati, SE	13:00 - 14:30	Ruang 308	30
18	Basic Wound Care	30	Prasetya Budi Utomo, S.Kom	13:00 - 14:30	Ruang 312	30
19	Bahasa Inggris 2	30	Guno Ariyanto	14:45 - 16:15	Ruang 308	30
20	Bahasa Inggris 2	21	Ira Budi Setyowati, SE	14:45 - 16:15	Ruang 306	30
21	Rabu, 29 Januari 2020					
21	Kepawatan Medikal Bedah 1	30	Guno Ariyanto	07:00 - 08:30	Ruang 311	30
22	Kepawatan Medikal Bedah 1	21	Nurul Niawati	07:00 - 08:30	Ruang 310	30
23	Kepawatan Keluarga	30	Ira Budi Setyowati, SE	08:45 - 10:15	Ruang 311	30
24	Kepawatan Keluarga	21	Prasetya Budi Utomo, S.Kom	08:45 - 10:15	Ruang 307	30
25	Kepawatan Maternitas	30	Surwaji Wibowo	10:30 - 12:00	Ruang 307	30
26	Kepawatan Maternitas	21	Guno Ariyanto	10:30 - 12:00	Ruang 311	30
27	Ilmu Biomedik Dasar	30	Nurul Niawati	13:00 - 14:30	Ruang 312	30
28	Ilmu Biomedik Dasar	17	Ira Budi Setyowati, SE	13:00 - 14:30	Ruang 306	30
29	Kepawatan Dasar	30	Nurul Niawati	14:45 - 16:15	Ruang 308	30
30	Kepawatan Dasar	17	Surwaji Wibowo	14:45 - 16:15	Ruang 310	30
31	Kamis, 30 Januari 2020					
31	Bahasa Inggris Keperawatan 2	1	Ira Budi Setyowati, SE	07:00 - 08:30	Ruang 305	30
32	Sistem Informasi Keperawatan	30	Surwaji Wibowo	08:45 - 10:15	Ruang 305	30
33	Sistem Informasi Keperawatan	18	Surwaji Wibowo	08:45 - 10:15	Ruang 308	30
34	Kepawatan HIV - AIDS	30	Ira Budi Setyowati, SE	10:30 - 12:00	Ruang 309	30
35	Kepawatan HIV - AIDS	18	Nurul Niawati	10:30 - 12:00	Ruang 311	30
36	Biostatistik	30	Surwaji Wibowo	13:00 - 14:30	Ruang 312	30

Gambar 5. 6 Hasil penjadwalan dengan jumlah kromosom 30 dan maksimal generasin 35.

Pengujian empat dilakukan menggunakan aplikasi dengan memasukkan data penjadwalan ujian akhir semester ganjil fakultas ilmu kesehatan ajaran 2019/2020. proses input dapat dilihat pada gamabar 5.7. Data pengujian menggunakan aplikasi adalah sebagai berikut :

The screenshot shows a web application titled "Algoritma Genetika Penjadwalan Ujian" for the Faculty of Health Sciences at Muhammadiyah Magelang. The main section is "Generate Jadwal". It features two input fields: "Jumlah Kromosom Dibangkitkan" with a value of 50, and "Maksimal Generasi" with a value of 100. Below these are buttons for "Generate Jadwal" and "Lihat Jadwal". A status bar at the bottom indicates "FITNESS TERBAIK: 1" and "GENERASI: 28". The footer contains the copyright information: "© 2020 Algoritma Genetika Penjadwalan Ujian, Fakultas Ilmu Kesehatan Universitas Muhammadiyah Magelang".

Gambar 5. 7 Hasil Jadwal Uji ke empat dengan Aplikasi Algoritma Genetika.

Hasil pengujian dilakukan dengan Jumlah data matakuliah 45, jumlah ruangan 8, jumlah hari 10, dan 5 pengawas dengan jumlah kromosom 50 dan banyak generasi 100 hasilnya belum optimal dengan nilai fitness 1. Hasil dari penjadwalan dapat dilihat pada gambar 5.8.

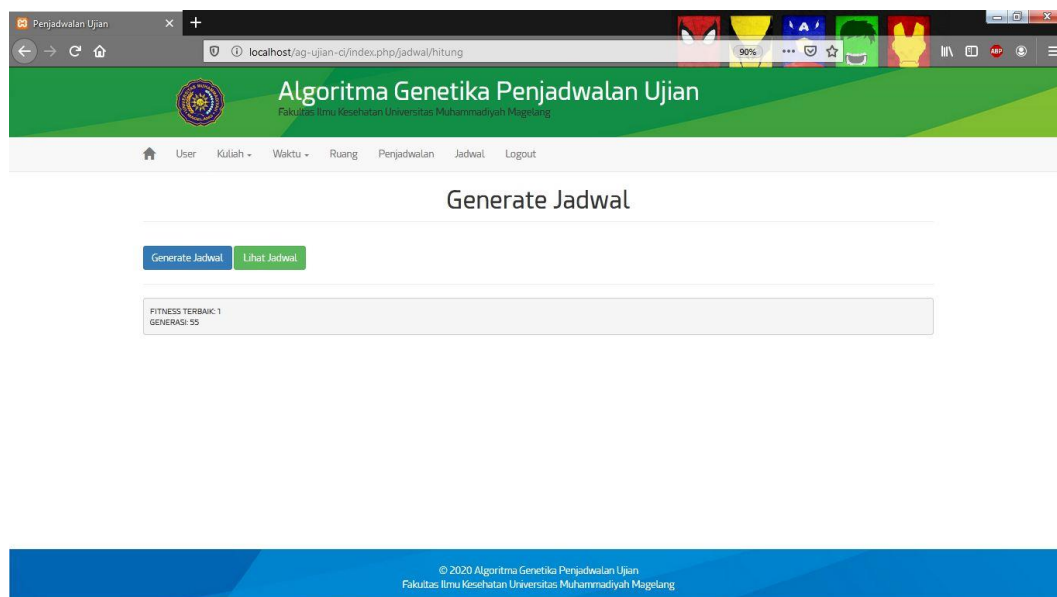
No	Hari, Tanggal	Mata Kuliah	Peserta	Pengawas	Jam	Ruang	Kapasitas
1	Senin, 27 Januari 2020	Psikologi	30	Ira Budi Setyowulan, SE	07.00 - 08.30	Ruang 312	30
2		Psikologi	17	Surwaji Wibowo	07.00 - 08.30	Ruang 305	30
3		Konsep Dasar Keperawatan	30	Nurul Niawati	08.45 - 10.15	Ruang 305	30
4		Konsep Dasar Keperawatan	17	Ira Budi Setyowulan, SE	08.45 - 10.15	Ruang 305	30
5		Gizi Dan Diet	30	Surwaji Wibowo	10.30 - 12.00	Ruang 311	30
6		Gizi Dan Diet	17	Prasetyo Budi Utomo, S.Kom	10.30 - 12.00	Ruang 305	30
7		Antropologi Kesehatan	30	Guno Ariyanto	13.00 - 14.30	Ruang 305	30
8		Antropologi Kesehatan	18	Ira Budi Setyowulan, SE	13.00 - 14.30	Ruang 308	30
9		Metodologi Keperawatan	30	Ira Budi Setyowulan, SE	14.45 - 16.15	Ruang 311	30
10		Metodologi Keperawatan	21	Guno Ariyanto	14.45 - 16.15	Ruang 306	30
11	Selasa, 28 Januari 2020	Manajemen patient Safety	30	Surwaji Wibowo	07.00 - 08.30	Ruang 309	30
12		Manajemen patient Safety	22	Prasetyo Budi Utomo, S.Kom	07.00 - 08.30	Ruang 307	30
13		Manajemen Keperawatan	30	Ira Budi Setyowulan, SE	08.45 - 10.15	Ruang 307	30
14		Manajemen Keperawatan	21	Nurul Niawati	08.45 - 10.15	Ruang 305	30
15		Bahasa Arab	30	Surwaji Wibowo	10.30 - 12.00	Ruang 305	30
16		Bahasa Arab	21	Ira Budi Setyowulan, SE	10.30 - 12.00	Ruang 311	30
17		Basic Wound Care	30	Ira Budi Setyowulan, SE	13.00 - 14.30	Ruang 305	30
18		Basic Wound Care	21	Surwaji Wibowo	13.00 - 14.30	Ruang 309	30
19		Bahasa Inggris 2	30	Prasetyo Budi Utomo, S.Kom	14.45 - 16.15	Ruang 309	30
20		Bahasa Inggris 2	21	Guno Ariyanto	14.45 - 16.15	Ruang 310	30

Gambar 5. 8 Hasil penjadwalan dengan jumlah kromosom 50 dan maksimal generasin 100.

Pada pengujian diatas dapat dilihat bahwa dengan memasukkan jumlah nilai kromosom 50 dan banyak generasi 100 dapat meghasilkan jadwal yang optimal sehingga tidak terjadi bentrokan jadwal antar pengawas.

Dari hasil pengujian diatas penulis memutuskan untuk mematenkan jumlah kromosom dibangkitkan = 50 dan generasi = 100 sebagai nilai tepat untuk mengisi indikator parameter, dan menghilangkan indikator parameter memasukkan jumlah kromosom yang dibangkitkan dan maksimal generasi sehingga memudahkan admin untuk melakukan

proses penjadwalan. Tampilan penjadwalan yang baru dapat dilihat pada gambar 5.9.



Gambar 5.9 Tampilan Penjadwalan setelah dihilangkan indikator parameter.

Dari gambar 5.9 dapat kita lihat dengan menekan tombol generate jadwal dapat menghasilkan jadwal yang optimal dengan nilai fitness adalah 1 yang artinya pada jadwal tidak terjadi bentrokan pengawas.

2. Hasil Pengujian Black Box

Pengujian dengan Black Box yaitu pengujian fungsional tanpa melihat alur eksekusi program, namun cukup dengan memperhatikan apakah setiap fungsi sudah berjalan dengan sesuai fungsi yang diuji. Adapun yang diuji meliputi adalah proses login, tombol-tombol pada menu bar, tombol tambah, simpan, edit, hapus, generate dan cetak.

3. Hasil Acceptance Testing

Pengujian sistem ini dilakukan dengan penyebaran kuisioner yang diberikan pada 3 responden selaku petugas pembuat jadwal ujian. Adapun setelah penyebaran kuisioner tersebut selanjutnya dilakukan rekap

terhadap hasil kuisioner yang telah disebar dengan menggunakan perhitungan skala likert. Pengujian skala likert digunakan sebagai suatu perhitungan untuk mengolah informasi yang didapat dari hasil kuesioner menjadi suatu informasi. Pengujian skala likert digunakan karena memiliki banyak kemudahan dalam menggukannya dan mempunyai reliabilitas tinggi (Rahadi, 2014). Berdasarkan persentase hasil *satisfication testing* di atas, maka diperoleh Rekap Nilai *Satisfication*. Adapun langkah – langkah untuk mengolah data kuesioner sebagai berikut.

a. Penentuan Skor

Tabel 5. 1 Tabel Penentuan Skor

Skala Jawaban	Keterangan	Nilai
STS	Sangat Tidak Setuju	1
TS	Tidak Setuju	2
N	Netral	3
S	Setuju	4
SS	Sangat Setuju	5

Setelah itu, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = \frac{TS}{Skor\ Ideal} \times 100 \%$$

Dimana:

Y = Nilai Persentase

TS = Total Skor Responden

Skor Ideal = Skor tertinggi x jumlah responden

Sebelum menyelesaikannya kita juga harus mengetahui interval (rentang jarak) dan interpretasi persen agar mengetahui penilaian dengan metode mencari Interval skor persen (I).

Rumus Interval

$I = 100 / \text{Jumlah Skor (Likert)}$

Maka $= 100 / 5 = 20$

Hasil (I) = 20

(Ini adalah intervalnya jarak dari terendah 0 % hingga tertinggi 100%)

Tabel 5. 2 Tabel Presentase Kriteria Skor

Keterangan	Kategori
Sangat Tidak Setuju	0 % - 19,99 %
Kurang Setuju	20 % - 39,99 %
Netral	40 % - 59,99 %
Setuju	60 % - 79,99 %
Sangat Setuju	80 % - 100 %

Berikut dibawah ini pada tabel 5.3 adalah hasil presentase masing-masing jawaban yang sudah dihitung nilainya. Kuesioner ini telah diujikan kepada 3 orang responden.

Tabel 5. 3 Tabel Hasil Kueisioner

No	Pertanyaan	SKALA					Jumlah Responden	SKOR					Total	Skor ideal	Persen	katergori
		STS	TS	N	S	SS		STS*1	TS*2	N*3	S*4	SS*5				
1	Apakah menu atau fitur yang disediakan oleh sistem ini mudah digunakan?	0	0	1	2	0	3	0	0	3	8	0	11	15	73	Setuju
2	Apakah tata letak menu serta isi pada tiap – tiap menu dalam sistem sudah sesuai?	0	0	2	1	0		0	0	6	4	0	10	15	66	Setuju
3	Apakah menu atau fitur dalam sistem ini berjalan dengan baik?	0	0	0	3	0		0	0	0	12	0	12	15	80	Setuju
4	Apakah sistem ini mudah dioperasikan atau dijalankan?	0	0	1	2	0		0	0	3	8	0	11	15	73	Setuju
5	Secara sistem ini telah dapat diakses sesuai dengan hak aksesnya masing – masing?	0	0	3	0	0		0	0	9	0	0	9	15	60	Setuju
6	Apakah sistem ini sesuai dengan kebutuhan?	0	0	1	2	0		0	0	3	8	0	11	15	73	Setuju
7	Apakah sistem ini bermanfaat bagi pengguna?	0	0	0	2	1		0	0	0	8	5	13	15	86	Sangat setuju
8	Apakah sistem ini memiliki kemampuan dan fungsi sesuai yang diharapkan?	0	0	2	1	0		0	0	6	4	0	10	15	66	Setuju
9	Apakah output berupa laporan sudah sesuai yang diharapkan? (Output data berupa pdf)	0	0	0	3	0		0	0	0	12	0	12	15	80	Setuju
Jumlah							0	0	30	64	5	99	135	657		
Total													657			
Rata-Rata													73	Setuju		

Dari hasil kuisisioner diatas dapat disimpulkan bahwa 73% dari 3 reponden sudah setuju dengan fungsi dan fitur yang disediakan pada sistem. Fitur dan menu yang disediakan dalam sistem juga sangat mudah dioperasikan oleh pengguna dan sudah sesuai dengan kebutuhan.

B. Pembahasan

Otomatisasi sistem penjadwalan ujian akhir semester menggunakan algoritma genetika yang digunakan oleh staff Tata Usaha, untuk pembuatan jadwal ujian akhir semester di Fakultas Ilmu Kesehatan Muhammadiyah Universitas Muhammadiyah Magelang. Dari hasil kuisisioner yang diberikan pada 3 responden, 73 % menyatakan sistem sudah sesuai dengan kebutuhan serta fitur yang disediakan pada sistem mudah digunakan, tata letak menu sudah sesuai dan mudah dioperasikan. Kemudian 27 % responden menyatakan sistem belum sesuai dengan hak aksesnya serta output yang disediakan belum sesuai dengan yang diharapkan.

Sistem ini dibangun disesuaikan dengan perancangan sistem yang sudah dibuat. Desain antarmuka dan basis data yang digunakan juga sudah disesuaikan dengan kebutuhan pada sistem.

1. Kelebihan Sistem

Kelebihan Algoritma Genetika Pada Penjadwalan Ujian Akhir Semester adalah sebagai berikut :

- a. Penjadwalan dapat ditentukan dengan lebih cepat.
- b. Lebih terstruktur dalam pengaturan Jadwal.
- c. Mempermudah menjadwalkan pengawas ujian dengan keterbatasan jumlah pengawas.

2. Manfaat Sistem

Manfaat yang diperoleh dari sistem penjadwalan ujian akhir semester fakultas ilmu kesehatan ummgl menggunakan metode algoritma genetika adalah sebagai berikut :

- a. Memudahkan kaprodi dalam pembuatan jadwal ujian.

- b. Memudahkan kaprodi dalam menentukan ruang pelaksanaan ujian.
- c. Kaprodi dapat lebih cepat menentukan jadwal pengawasan ujian.
- d. Kaprodi dapat menghindarkan terjadinya tabrakan jadwal antara ruang ujian dengan pengawas ujian.

BAB VI

KESIMPULAN DAN SARAN

Setelah melalui proses analisis perencanaan, implementasi, serta pengujian maka pada bab ini akan dibahas kesimpulan tentang hasil. Selain kesimpulan dari permasalahan yang diangkat juga akan disampaikan saran-saran yang dapat memberikan masukan dan catatan-catatan guna pengembangan sistem menjadi yang lebih baik.

A. KESIMPULAN

Dari hasil penelitian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Telah berhasil dirancang solusi penjadwalan ujian otomatis dua Prodi secara bersamaan dan menjamin seorang pengawas tidak mengawasi ujian dalam waktu yang bersamaan dengan menggunakan Algoritma Genetika
2. Algoritma genetika mampu menyelesaikan permasalahan penjadwalan ujian dua prodi secara bersamaan di mana algoritma genetika memperlihatkan hasil fitness akhir dengan nilai 1 yang mana menjadi solusi terbaik yang berarti tidak terjadi bentrokan jadwal pengawas.

B. SARAN

Sistem penjadwalan ini mungkin masih dapat dikembangkan lagi dengan menambahkan beberapa fungsi atau menu tambahan dan juga tampilan atau desain antarmuka dapat dibuat lebih menarik lagi, Penambahan tenaga pengawas juga sangat diperlukan agar keakuratan sistem dapat lebih meningkat dikarenakan semakin banyaknya jumlah pengawas mempengaruhi dengan tingkat keakuratan sistem dalam membuat ujian akhir semester.

DAFTAR PUSTAKA

- Aditya, L.A., & Mega W., 2017. Implementasi Algoritma Genetika untuk Penjadwalan Mata Pelajaran pada LMS GETSMART. Matik Penusa : Jurnal Manajemen dan Informasi Pelita Nusantara, Volume 21, Nomor 1, Juni 2017, halaman 65-70.
- Arifudin, Riza. 2011. Optimasi Penjadwalan Proyek dengan Penyeimbangan Biaya Menggunakan Kombinasi CPM dan Algoritma Genetika. Jurnal Masyarakat Informatika, Volume 2, Nomor 4, halaman 1-14.
- Arkeman, Kundang dan Hendra. 2012. *Algoritma Genetika Teori dan Aplikasinya untuk Bisnis dan Industri*. IPB Press, Bandung.

- Baker, Kenneth R.(1974), *Introduction To Sequencing and Scheduling*, Jhon Willey and Sons, Inc. New York.
- Fadlisyah, Arnawan, Faisal. 2009. *Algoritma Genetika*. Graha Ilmu, Yogyakarta.
- Farida, N. I. 2008. *Sistem Pendukung Keputusan Penjadwalan Pengajar Praktikum Laboratorium Komputer STIKOM Menggunakan Algoritma Genetika*. STIKOM Surabaya, Surabaya.
- Gata Windu, Gata Grace, 2013. *Sukses Membangun Aplikasi Penjualan dengan Java*. PT. Elex Media Komputindo Kelompok Gramedia, Anggota IKAPI, Jakarta 2013.
- Gen, M. & R. Cheng. 2000. *Genetic Algorithm and Engineering Optimization*. Jhon Wiley and Sons. Inc. Newyork.
- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. United State of America : Addison-Westley.
- Haupt, R. L. & Haupt. 2004. *Practical Genetic Algorithmns*. New Yersey: Jhon Wiley dan Sons,Inc.
- Ivan, Stephanus Raphael dan Halim Agung. 2018. Aplikasi Penjadwalan Mata Pelajaran di SMAN 31 Menggunakan Algoritma Genetika Berbasis WEB. Jurnal SIMETRIS, Volume 9, Nomor 1, April 2018, Halaman 641-656.
- Maghfira, Tusty Nadia dkk. Optimasi Penjadwalan Pengajar Menggunakan Algoritma Genetika Pada Royal English: Toefl & Toeic Center Malang. Skripsi Sarjana, Fakultas Ilmu komputer Universitas Brawijaya Malang, malang, 2016.

- Maharani Febria. Sistem Penjadwalan Proyek Menggunakan Algoritma Genetika. Skripsi Sarjana, Fakultas Sains dan Teknologi Universitas Islam Negeri SUSKA, Riau, 2013
- Pradnyana, Nanda Bagus, Dwi Sunaryono, dan Abdul Munif. 2012. Perancangan dan Pembuatan Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetik dan Teknologi Java API for XML Web Service pada Platform Android. *Jurnal Teknik Pomits Volume 1, Nomor 1, (2012) 1-5.*
- Putra, Y. Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika. Skripsi Sarjana, Fakultas Sains dan Teknologi Universitas Islam Negeri SUSKA, Riau, 2009
- Rahadi, D. R. (2014). Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android Interface pengguna Android didasarkan pada manipulasi langsung menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata , seperti menggesek (swiping), mengetuk . 6(1), 661–671.
- Sitanggang, D. 2015. Inisialisasi Populasi Pada Algoritma Genetika Menggunakan Simple Hill Climbing (SHC) Untuk Traveling Salesman Problem (TSP). *Jurnal Times, Volume 4, Nomer 2, Halaman 40-44.*
- Suyanto. 2005. *Algoritma Genetika Dalam MATLAB*. Andi. Yogyakarta
- Toscany, A., N., & Rusdianto, R. 2017. Penegembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetika. *Jurnal Manajemen Sistem Informasi, Volume 2, Nomor 2, Juni 2017, Halaman 379-393.*