

PENERAPAN ALGORITMA GENETIKA DALAM MENGATASI JADWAL MENGAJAR YANG BENTROK PADA PROGRAM STUDI INFORMATIKA IBI KOSGORO 1957 JAKARTA INDONESIA

PENULIS

¹⁾Boy Firmansyah, ²⁾Dwi Sidik Permana, ³⁾Natalia Evianti, ⁴⁾Asep Mulyana Wihandar, ⁵⁾Ari Kurniawan

ABSTRAK

Menjadwalkan jadwal mengajar di kampus dilakukan untuk mendukung, mempercepat, dan meningkatkan kualitas kampus. Penjadwalan mata kuliah pada umumnya berfungsi sebagai kegiatan dalam meningkatkan kualitas pengajaran dosen dan mahasiswa dalam melihat kedisiplinan dalam kegiatan kinerja kampus. Seiring dengan bertambahnya jadwal perkuliahan yang menuntut ketelitian secara tepat dan cepat dalam pengolahan datanya juga menjadikan efisien waktu. Selama ini pengelolaan data penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta masih sering terjadi ketidaksesuaian jadwal sampai terjadinya jadwal yang bentrok karena cukup banyak jadwal kesediaan mengajar dosen yang tidak terukur. Untuk itu dirasa perlu untuk merancang aplikasi penjadwalan mata kuliah yang dirancang agar berfungsi untuk membantu Ketua Program Studi dalam penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta dengan menggunakan PHP dan MYSQL. Algoritma genetika dipilih sebagai pemrosesan jadwal karena diharapkan dapat membantu proses penjadwalan mata kuliah Ketika admin memasukkan jadwal sesuai tahun ajaran kemudian diolah untuk memastikan tidak ada jadwal yang bentrok.

Kata Kunci

Penjadwalan, Algoritma Genetika, PHP, MySQL, Prodi Informatika IBI Kosgoro 1957

AFILIASI

Prodi, Fakultas
Nama Institusi
Alamat Institusi

¹⁾⁻⁵⁾Program Studi Teknik Informatika, Fakultas Ilmu Komputer
¹⁾⁻⁵⁾Institut Bisnis dan Informatika (IBI) Kosgoro 1957
¹⁾⁻⁵⁾Jl. M. Kahfi II No. 33, Jagakarsa, Jakarta Selatan, DKI Jakarta

KORESPONDENSI

Penulis
Email

Boy Firmansyah
boy@ibi-k57.ac.id

LICENSE



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

I. PENDAHULUAN

Perkembangan teknologi saat ini sangat pesat yang mempengaruhi lingkungan sekitar, seakan-akan segala aktivitas manusia tidak lepas dari teknologi. Dengan keberadaan komputer dari masa ke masa yang semakin canggih maka sangat membantu untuk mewujudkan segala hal yang dibutuhkan oleh manusia. Berkat teknologi ini, semuanya dapat terwujud secara efektif dan efisien.

Menjadwalkan jadwal di kampus untuk mendukung, mempercepat, dan meningkatkan kualitas kampus (Janata, Ari, Haerani: 2015). Penjadwalan mata kuliah pada umumnya berfungsi sebagai kegiatan dalam meningkatkan kualitas pengajaran dosen dan mahasiswa dalam melihat kedisiplinan dalam kegiatan kinerja kampus. Seiring dengan bertambahnya jadwal perkuliahan yang menuntut ketelitian secara tepat dan cepat dalam pengolahan datanya juga menjadikan efisien waktu (Puspaningrum, Ayu, Djunaidy, Vinarti: 2013).

Selama ini pengelolaan data penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta masih dilakukan secara manual dan sering terjadi ketidaksesuaian jadwal sampai terjadinya jadwal yang bentrok karena cukup banyak jadwal kesediaan mengajar dosen yang tidak terukur sehingga diperlukan perancangan sistem agar semuanya dapat diselesaikan dengan baik.

Berdasarkan uraian di atas, peneliti akan merancang aplikasi penjadwalan mata kuliah yang dirancang dengan menggunakan PHP dan MYSQL. Aplikasi yang dirancang berfungsi untuk membantu Ketua Program Studi dalam penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta.

Menjadwalkan kuliah di kampus adalah untuk mendukung, mempercepat, dan meningkatkan kualitas kampus. Penjadwalan mata kuliah pada umumnya berfungsi sebagai kegiatan dalam meningkatkan kualitas pengajar dan mahasiswa dalam melihat kedisiplinan dalam kegiatan kinerja kampus. Seiring dengan bertambahnya jadwal perkuliahan yang menuntut ketelitian secara tepat dan cepat dalam pengolahan datanya juga menjadikan efisien waktu.

Selama ini pengelolaan data penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta masih dilakukan secara manual dan sering terjadi ketidaksesuaian jadwal karena cukup banyak kelas dan dosen yang berhalangan hadir dan juga adanya masalah jadwal yang bentrok sehingga diperlukan perancangan sistem agar semuanya dapat diselesaikan dengan baik.

Berdasarkan uraian di atas, peneliti akan merancang aplikasi penjadwalan mata kuliah yang dirancang dengan menggunakan PHP dan MySQL dengan menggunakan Algoritma Genetika. Aplikasi yang dirancang berfungsi untuk membantu Ketua Program Studi dalam penjadwalan mata kuliah di IBI Kosgoro 1957 Jakarta.

II. METODE PENELITIAN

Adapun penelitian ini dilaksanakan dengan memakai pendekatan penelitian kualitatif. Peneliti melaksanakan survei di IBI Kosgoro 1957 yang menjadi objek dari penelitian lalu melaksanakan analisis terhadap Penjadwalan Mengajar Dosen. Analisis ini meliputi observasi faktor-faktor yang mempengaruhi kinerja penjadwalan yaitu banyaknya mata kuliah, banyaknya dosen, banyaknya kelas termasuk kelas paralel, jumlah hari berikut sesi mengajar dan artikel pengabdian kepada masyarakat, serta jumlah ruangan yang tersedia.

2.1 Populasi, Sampel, dan Teknik Pengambilan Sampel

Dalam penelitian ini seluruh dosen tetap dan dosen tidak tetap yang mengajar pada program studi Informatika akan menjadi sampel total. Jadi pada penelitian ini diambil sampel dari jadwal mengajar semester ganjil tahun ajaran 2020/2021 program studi Informatika Fakultas Ilmu Komputer IBI Kosgoro 1957.

2.2 Pengumpulan Data dan Instrumen Penelitian

Adapun pada penelitian ini, instrumen yang digunakan dalam pengumpulan data adalah seluruh data jadwal dosen yang telah dibuat sebelumnya oleh Ketua Program Studi Informatika.

2.3 Analisis Data

Kegiatan analisis data di dalam penelitian ini dilakukan secara bersamaan dengan kegiatan pengumpulan data di lapangan. Dan dari data yang terkumpul berikutnya dianalisis dengan menggunakan cara:

- 1) Reduksi data
- 2) Menampilkan (*display*) data
- 3) Membuat kesimpulan serta verifikasi. (Salhi, Saïd: 2017)

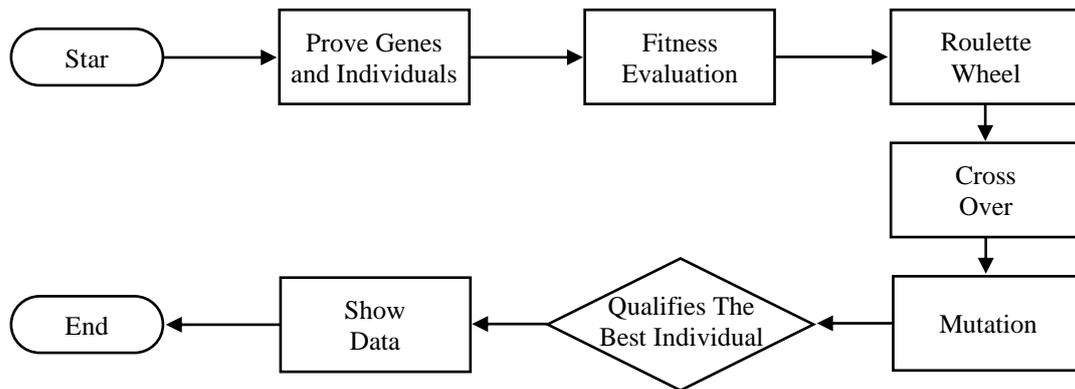
III. HASIL DAN PEMBAHASAN

Penjadwalan berkaitan dengan kegiatan dalam hal membuat jadwal. Jadwal adalah tabel kegiatan yang diatur berdasarkan waktu penempatannya (Jain, Ashish, Jain, Chande: 2010). Kegiatan ini biasanya merupakan pertemuan antara beberapa komponen pada waktu dan tempat yang sama. Jadwal harus memenuhi beberapa persyaratan dan paling sesuai dengan keinginan semua orang yang terlibat. Pengaturan waktu kegiatan harus diatur agar tidak ada satu komponen pun yang memiliki lebih dari satu kegiatan pada waktu yang sama (Widodo, Wahyu, Mahmudy: 2010). Penjadwalan kegiatan belajar mengajar merupakan suatu pengaturan perencanaan belajar mengajar yang meliputi mata kuliah, dosen, waktu dan tempat di kampus. Secara umum penjadwalan kegiatan belajar mengajar disajikan dalam tabel hari dalam seminggu yang terdiri dari *slot* waktu yang terdiri dari mata kuliah, hari, jam, dan dosen sesuai dengan mata kuliah yang diajarkan (Hamad, Safwat A., Omara: 2016).

Dalam ilmu komputer dan matematika, definisi dari algoritma yaitu urutan dari beberapa langkah logis dan sistematis yang dipakai untuk menyelesaikan suatu masalah tertentu. Yang lain berpendapat bahwa definisi algoritma adalah proses atau sekumpulan aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah lainnya, terutama oleh komputer (Hermawanto, Denny: 2014). Dengan kata lain, semua tatanan logika yang disusun berdasarkan sistematika tertentu dan dipakai dalam rangka menyelesaikan suatu masalah dapat disebut algoritma (Elsayed, Saber M., Sarker, Essam: 2014).

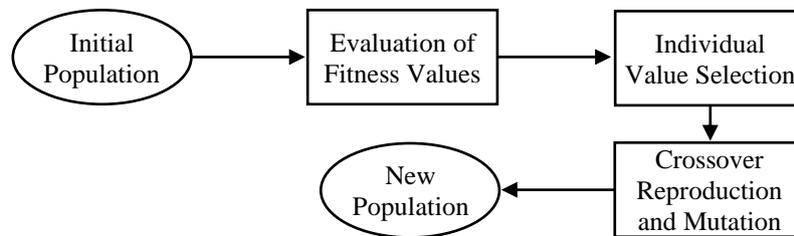
Algoritma digunakan untuk melakukan perhitungan, penalaran otomatis, dan mengolah data di komputer menggunakan *software* (Kumar, Manoj, et al: 2010). Dalam algoritma terdapat rangkaian instruksi yang terbatas untuk menghitung suatu fungsi yang jika dijalankan dan diproses akan menghasilkan keluaran, kemudian berhenti pada kondisi akhir yang telah ditentukan (Ashari, Ahmad: 2016).

Menurut Donald Ervin Knuth, definisi algoritme adalah sekumpulan aturan hingga yang menyediakan rangkaian operasi untuk menyelesaikan masalah tertentu. Algoritma Genetika didasarkan pada proses genetik yang ada pada makhluk hidup, yaitu perkembangan generasi dalam suatu populasi alami, secara bertahap mengikuti prinsip seleksi alam. Dengan meniru teori evolusi ini, Algoritma Genetika dapat digunakan untuk mencari solusi atas permasalahan di dunia nyata. Pada awal perkembangannya, metode Algoritma Genetika ini pertama kalinya diperkenalkan oleh John Holland dari University of Michigan pada tahun 1975 dalam bukunya yang berjudul "*Adaption in Natural and Artificial Systems*", dan akhirnya dipopulerkan oleh salah satu mahasiswanya, David Goldberg, yang mampu memecahkan masalah sulit. yang melibatkan pengendalian transmisi pipa gas untuk disertasinya yang berjudul "*Computer-aided gas pipeline operation using genetic algorithms and rule learning*". Proses seleksi alam ini melibatkan perubahan gen yang terjadi pada individu melalui proses reproduksi (Rahman, F. W: 2011). Dalam algoritma genetika ini, proses pemuliaan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar pemikiran "Bagaimana cara mendapatkan keturunan yang lebih baik". Algoritma genetika ini ditemukan oleh John Holland dan dikembangkan oleh muridnya David Goldberg. Secara umum, algoritma genetika menyelesaikan masalah dengan aliran berikut:



Gambar 1. Penyempurnaan Algoritma Genetika Oleh David Goldberg

Untuk mengetahui apakah solusi individu baik atau tidak, setiap individu dalam populasi harus memiliki *fitness cost* (Sembiring, Martinelly: 2013). Melalui nilai perbandingan ini, solusi terbaik akan diperoleh dengan memilah-milah nilai perbandingan individu dalam populasi. Solusi terbaik ini akan dipertahankan, sedangkan solusi lain diubah untuk mendapatkan solusi lain, melalui tahapan *cross-over* dan mutasi. Sebelum menentukan jadwal kelas dilakukan dua kali pengecekan terlebih dahulu yaitu mencari hari dan jam yang masih kosong dan pengecekan prioritas yaitu pada hari dan jam mana yang paling diprioritaskan (Deep, Kusum, et al: 2009).



Gambar 2. Evaluasi Skor *Fitness* Oleh David Goldberg

Tabel 1. *Fitness Cost*. (Babaei, H., Karimpour, J., Hadidi, A.: 2015)

Aturan	Fitness Cost
Jadwal bertabrakan	Jumlah jadwal yang bentrok * 100%
Ruangan bertabrakan	Jumlah ruang yang bentrok * 100%
Kuliah lebih dari SKS nya	Jumlah mata kuliah semester * 100%
Jadwal yang bertumpuk	Jumlah jadwal bertumpuk * 100%

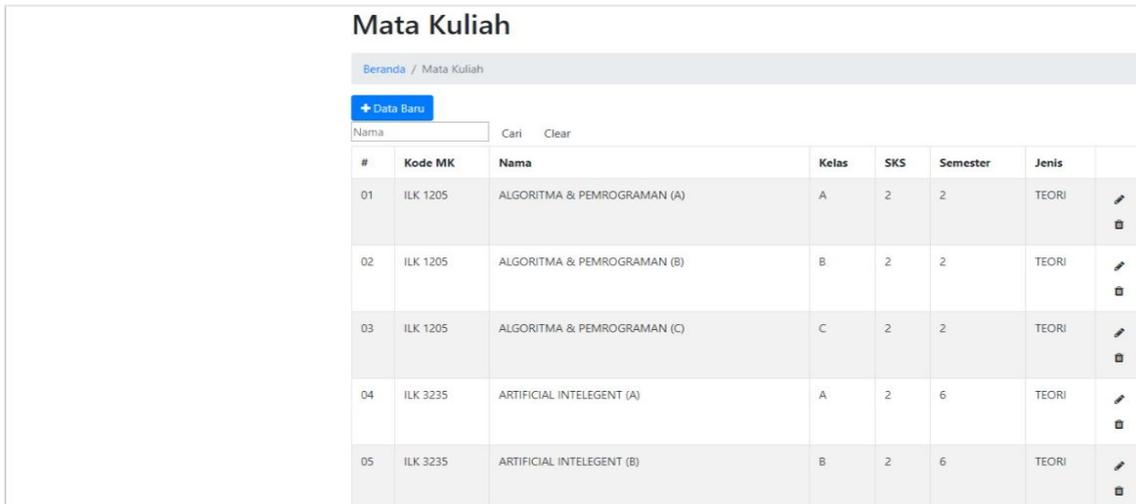
3.1 Halaman Penjadwalan

Halaman perencanaan berisi halaman untuk menghasilkan program untuk algoritma genetika. Halaman jadwal berisi entri parameter untuk algoritme genetika dan tombol untuk membuat jadwal dan mengunduhnya. Gambar 3 merupakan visualisasi halaman perencanaan.

Gambar 3. Form Penjadwalan

3.2 Halaman Mata Kuliah

Pada alaman mata kuliah ini adalah halaman yang berisikan informasi daripada mata kuliah yang berlangsung saat perkuliahan. Disini semua mata kuliah teori dan juga praktik digabung pada halaman ini. Dan di halaman ini pula *user* akan dapat melihat semua data mata kuliah, memperbaharui data-data mata kuliah, menambahkan data-data mata kuliah, menghapus data-data mata kuliah serta mencari nama-nama mata kuliah.

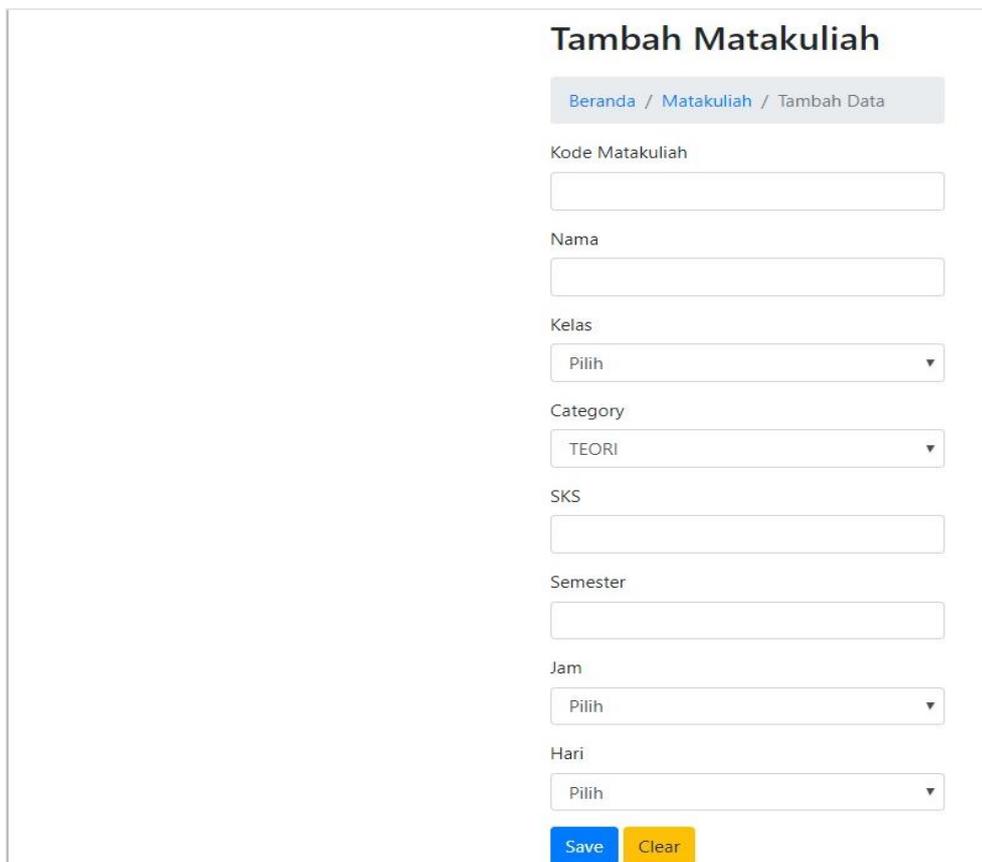


The screenshot shows a web interface titled "Mata Kuliah". It includes a breadcrumb "Beranda / Mata Kuliah", a "+ Data Baru" button, and a search bar. Below is a table with columns: #, Kode MK, Nama, Kelas, SKS, Semester, Jenis, and action icons (edit and delete).

#	Kode MK	Nama	Kelas	SKS	Semester	Jenis	
01	ILK 1205	ALGORITMA & PEMROGRAMAN (A)	A	2	2	TEORI	 
02	ILK 1205	ALGORITMA & PEMROGRAMAN (B)	B	2	2	TEORI	 
03	ILK 1205	ALGORITMA & PEMROGRAMAN (C)	C	2	2	TEORI	 
04	ILK 3235	ARTIFICIAL INTELEGENT (A)	A	2	6	TEORI	 
05	ILK 3235	ARTIFICIAL INTELEGENT (B)	B	2	6	TEORI	 

Gambar 4. Data Mata Kuliah

Sementara gambar 5 merupakan gambar dari halaman *form* menambahkan mata kuliah.



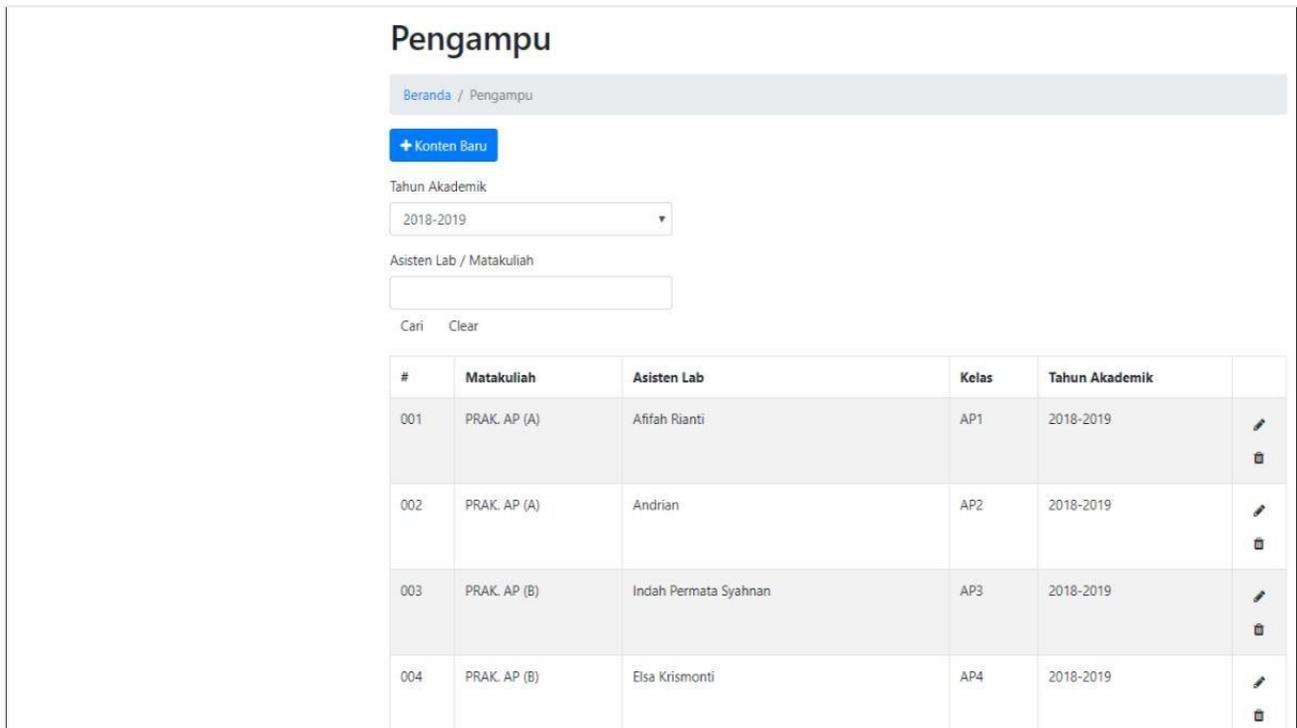
The screenshot shows a form titled "Tambah Matakuliah". It includes a breadcrumb "Beranda / Matakuliah / Tambah Data". The form fields are: Kode Matakuliah (text input), Nama (text input), Kelas (dropdown menu with "Pilih" selected), Category (dropdown menu with "TEORI" selected), SKS (text input), Semester (text input), Jam (dropdown menu with "Pilih" selected), and Hari (dropdown menu with "Pilih" selected). At the bottom are "Save" and "Clear" buttons.

Gambar 5. Form Tambah Mata Kuliah

Pada halaman ini, data hari, waktu, kelas, dan semester yang benar harus dimasukkan untuk kategori mata kuliah teori, karena hal ini mempengaruhi hasil pembuatan jadwal.

3.3 Halaman Pengampu

Halaman Pengampu (Dosen) berisikan halaman dari mata kuliah yang akan di *generate*. Dan di sini tersedia data-data ketua kelas dan mata kuliah yang akan diajar oleh pengampu tersebut. Gambar 6 adalah gambar halaman data pengampu.



Pengampu

Beranda / Pengampu

+ Konten Baru

Tahun Akademik
2018-2019

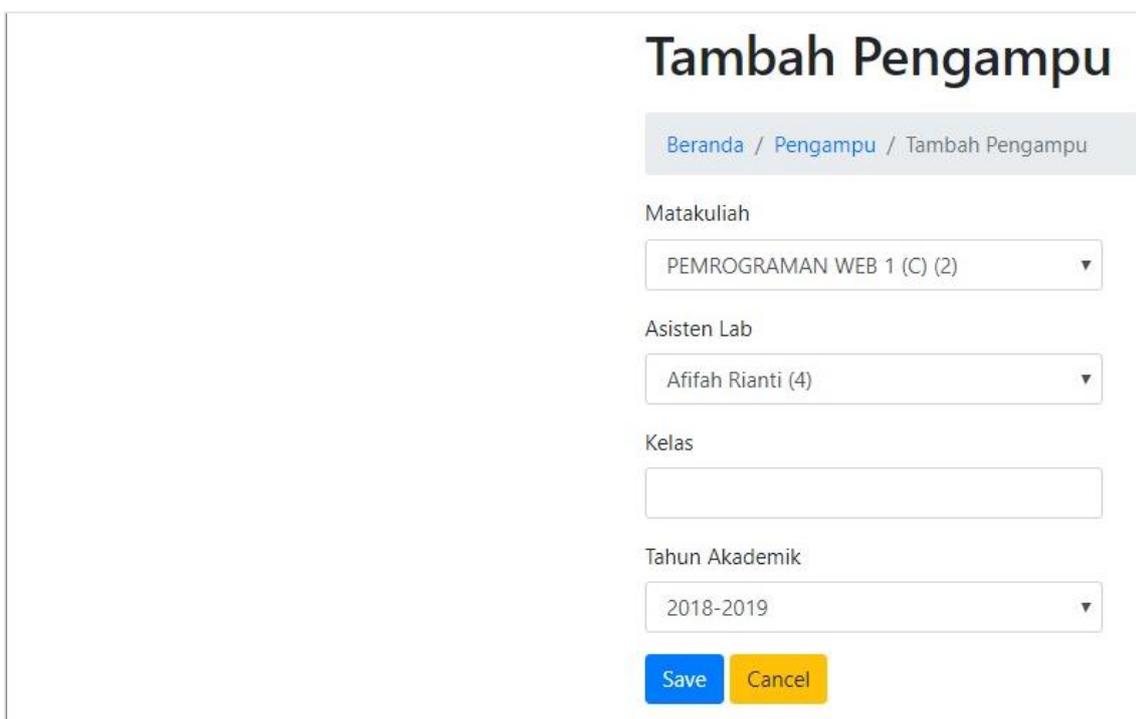
Asisten Lab / Matakuliah

Cari Clear

#	Matakuliah	Asisten Lab	Kelas	Tahun Akademik	
001	PRAK. AP (A)	Afifah Rianti	AP1	2018-2019	 
002	PRAK. AP (A)	Andrian	AP2	2018-2019	 
003	PRAK. AP (B)	Indah Permata Syahnun	AP3	2018-2019	 
004	PRAK. AP (B)	Elsa Krismonti	AP4	2018-2019	 

Gambar 6. Data Pengampu

Pada halaman ini *user* juga dapat melihat seluruh data pengampu, dan dapat pula memperbaharui data-data pengampu atau menghapus data dari pengampu. Dan juga dapat melakukan suatu pencarian data. Gambar 7 adalah gambar halaman *form* menambah pengampu.



Tambah Pengampu

Beranda / Pengampu / Tambah Pengampu

Matakuliah
PEMROGRAMAN WEB 1 (C) (2)

Asisten Lab
Afifah Rianti (4)

Kelas

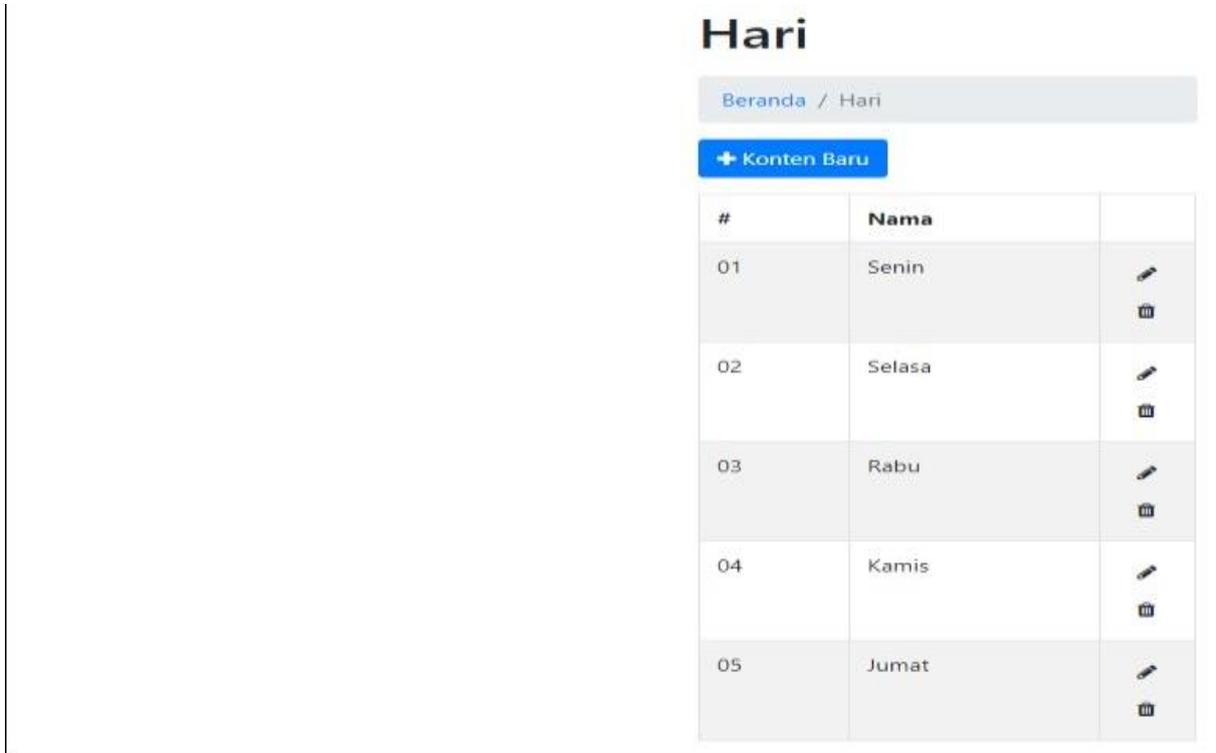
Tahun Akademik
2018-2019

Save Cancel

Gambar 7. Form Tambah Pengampu

3.4 Halaman Utilitas

Pada halaman utilitas akan berisikan data hari, kelas, ruang dan rentang jam. Untuk data bawaan pada hari terdiri dari rentang hari yang dimulai dari hari Senin sampai hari Jumat. Kemudian untuk rentang jam dapat beragam dan juga disesuaikan oleh waktu yang dikehendaki sementara bawaannya untuk kelas S1 Teknik Informatika umumnya terdiri dari 3 kelas, yaitu kelas A, kelas B dan kelas C atau didefinisikan juga sebagai kelas R1, R2 dan R3. Gambar 8 merupakan gambar halaman data utilitas hari

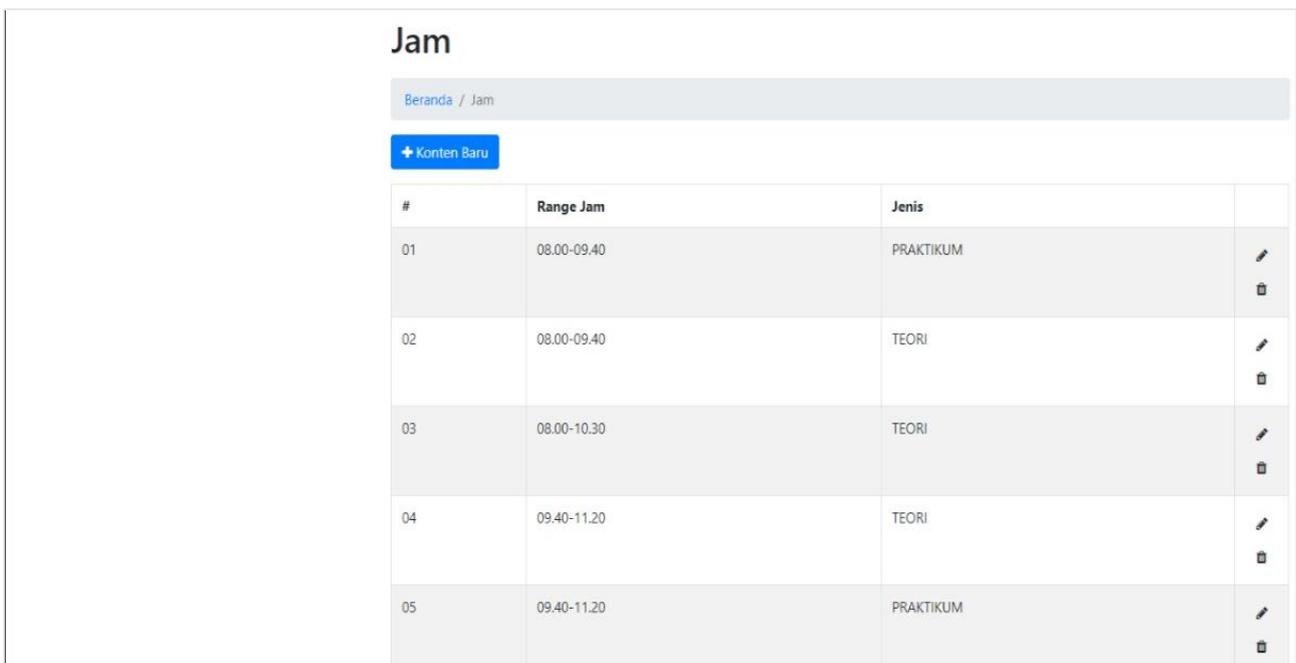


The screenshot shows a web interface for the 'Hari' utility. At the top, there is a breadcrumb 'Beranda / Hari' and a blue button '+ Konten Baru'. Below this is a table with three columns: '#', 'Nama', and an action column. The table contains five rows representing the days of the week from Monday to Friday.

#	Nama	
01	Senin	 
02	Selasa	 
03	Rabu	 
04	Kamis	 
05	Jumat	 

Gambar 8. Data Utilitas Hari

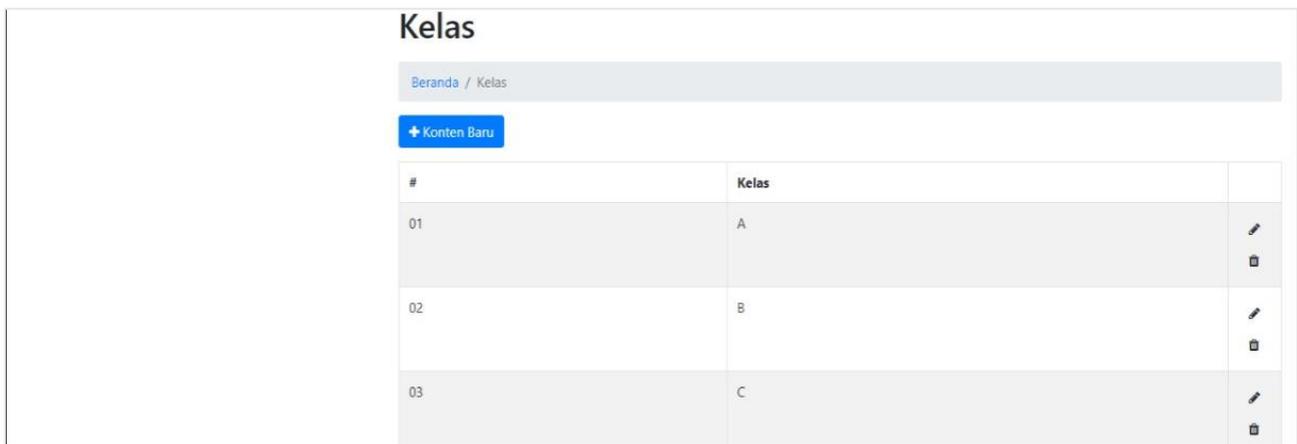
Sementara gambar 9 adalah gambar halaman dari data utilitas jam serta gambar 10 adalah gambar halaman dari data utilitas kelas



The screenshot shows a web interface for the 'Jam' utility. At the top, there is a breadcrumb 'Beranda / Jam' and a blue button '+ Konten Baru'. Below this is a table with four columns: '#', 'Range Jam', 'Jenis', and an action column. The table contains five rows representing different time slots and their types.

#	Range Jam	Jenis	
01	08.00-09.40	PRAKTIKUM	 
02	08.00-09.40	TEORI	 
03	08.00-10.30	TEORI	 
04	09.40-11.20	TEORI	 
05	09.40-11.20	PRAKTIKUM	 

Gambar 9. Data Utilitas Jam



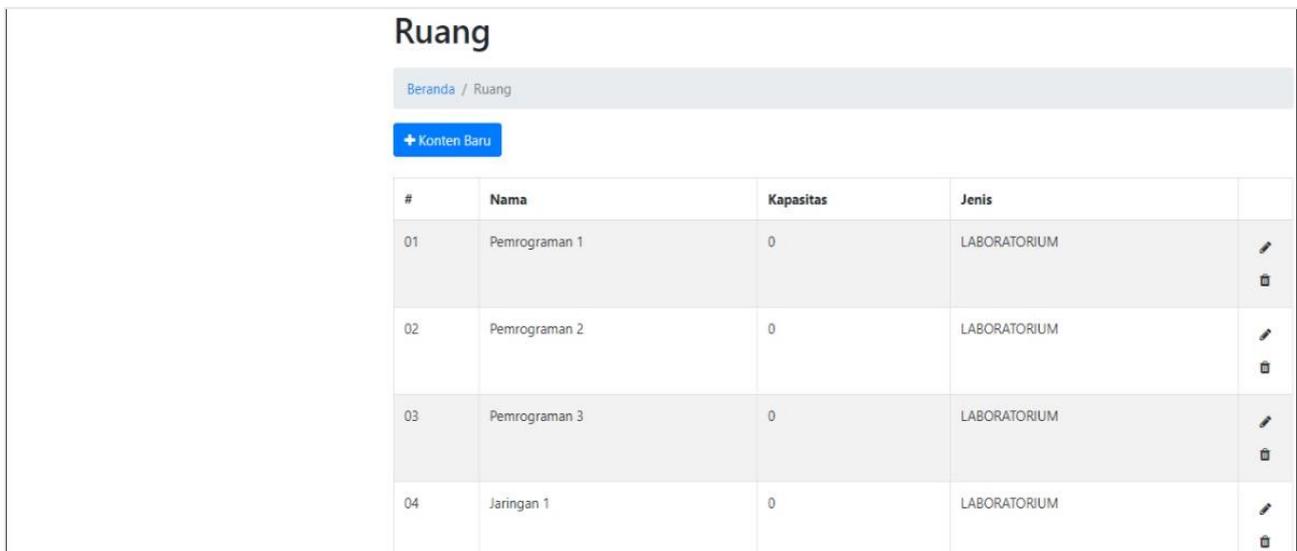
Kelas

Beranda / Kelas

+ Konten Baru

#	Kelas	
01	A	 
02	B	 
03	C	 

Gambar 10. Data Utilitas Kelas



Ruang

Beranda / Ruang

+ Konten Baru

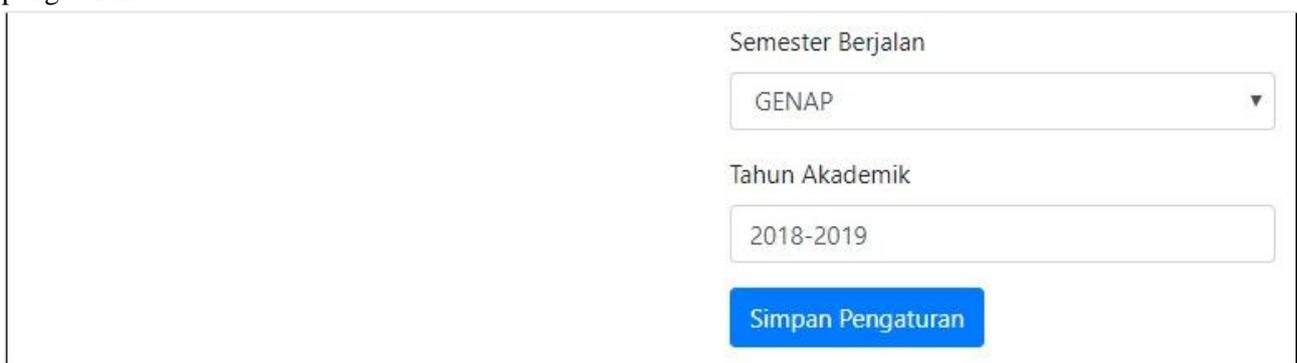
#	Nama	Kapasitas	Jenis	
01	Pemrograman 1	0	LABORATORIUM	 
02	Pemrograman 2	0	LABORATORIUM	 
03	Pemrograman 3	0	LABORATORIUM	 
04	Jaringan 1	0	LABORATORIUM	 

Gambar 11. Data Utilitas Ruang

Gambar 11 merupakan gambar data utilitas ruang. Di halaman utilitas ruang akan berisi nama ruangan yang akan digunakan sebagai proses pembelajaran. Yang mana umumnya berisikan nama ruangan kuliah yang digunakan untuk mengajar untuk pembelajaran teori maupun pembelajaran praktikum.

3.5 Halaman Pengaturan

Pada halaman pengaturan dijadikan sebagai halaman yang akan digunakan dalam pengolahan data yang akan digunakan. Dimana dengan mengatur data pengaturan pada halaman tersebut. Program hanya akan memuat data yang sesuai dengan pengaturan yang telah diisi. Gambar 12 merupakan gambar halaman *form* pengaturan.



Semester Berjalan

GENAP

Tahun Akademik

2018-2019

Simpan Pengaturan

Gambar 12. Form Pengaturan

Dengan melakukan pengaturan semester berjalan yaitu tahun akademik 2018-2019 semester genap maka hanya data-data tersebut yang akan ditampilkan serta hanya data tersebutlah yang akan dilakukan *generate* jadwal.

3.6 Pengujian

Pengujian dalam penelitian ini dikerjakan dengan membuktikan bahwasanya sistem ini telah bekerja dengan baik sesuai analisis dan perancangan sistem yang dilakukan sebelum membangun sistem. Dan untuk membuktikan bahwa sistem berjalan dengan baik, maka terdapat beberapa tahapan didalam pengujian, yaitu:

1) Tahap Sebelum *Generate*

Disini, halaman awal tampilannya seperti gambar 3 sebelum akan melakukan *generate* jadwal. Tapi telah ditampilkan hasil dari kegiatan *generate* jadwal dikarenakan peneliti telah menguji program beberapa kali serta operasi *generate* yang terakhir dilakukan otomatis tersimpan di tampilan awal. Gambar 13 adalah tampilan halaman sebelum kegiatan *generate* jadwal.

The screenshot shows a web interface titled "Penjadwalan". At the top, there is a breadcrumb "Beranda / Penjadwalan" and two buttons: "Proses" (blue) and "Export to Excel" (green). Below these are four input fields for parameters: "Jumlah Populasi" (10), "Probabilitas CrossOver" (0.70), "Probabilitas Mutasi" (0.40), and "Jumlah Generasi" (10000). At the bottom, there is a table with 8 columns: #, Hari, Jam, Matakuliah, SKS, Kelas, Asisten Lab, and Ruang. The table contains 5 rows of schedule data.

#	Hari	Jam	Matakuliah	SKS	Kelas	Asisten Lab	Ruang
1	Senin	08.00-09.40	PRAK. OAK (C) (4)	1	OAK5	Adam Yosafat Noverico Damanik (6) (B)	Jaringan 2
2	Senin	08.00-09.40	PRAK. PV (B) (4)	1	PV4	Mirta Amalia (6) (C)	Pemrograman 2
3	Senin	08.00-09.40	PRAK. MK (6)	1	MK1	Dormen Hutagalung (8) (C)	Multimedia
4	Senin	08.00-09.40	PRAK. TEKKOM (B) (6)	1	TEKKOM4	Cindy Laurent Ginting (8) (B)	Pemrograman 4
5	Senin	09.40-11.20	PRAK. OAK (B) (4)	1	OAK4	Zikri Akmal Santoso (8) (A)	Database

Gambar 13. Halaman Sebelum *Generate* Jadwal

Dan seperti yang terlihat di atas telah ada tampilan dari data hasil *generate* yang sebelumnya. Namun disini data tersebut dapat berubah seandainya sewaktu waktu ketika dikerjakannya *generate*. Kemudian tahap selanjutnya adalah tahap *generate* dari jadwal.

2) Tahap Melakukan *Generate*

Disini *user* akan diminta melakukan kegiatan *generate* jadwal dengan cara menekan tombol proses yaitu button yang berwarna biru. *User* akan diminta memasukkan data dari parameter yaitu jumlah populasi, jumlah generasi dan probabilitas *crossover* probabilitas *mutasi*. Jumlah populasi adalah jumlah awal dari kromosom yang akan dibangkitkan. Disini contohnya adalah jika *user* memasukkan suatu data jumlah populasi yaitu sebanyak 5, maka akan terdapat 5 jadwal yang mana merupakan solusi tapi hanya dipilih 1 jadwal yang merupakan jadwal terbaik. Probabilitas *crossover* adalah kemungkinan persilangan yang dapat terjadi antar data kromosom. Disini probabilitas mutasi merupakan suatu kemungkinan data yang mana akan ditukar. Jumlah generasi yaitu sebanyak apa suatu

proses genetika yang akan berjalan. Gambar 14. adalah gambar setelah *user* melaksanakan *generate* jadwal.

#	Hari	Jam	Matakuliah	SKS	Kelas	Ketua Kelas	Ruang
1	Senin	08.00-09.40	PRAK. SI (C) (4)	1	SI6	Mutiah Khairani (6) (A)	Jaringan 1
2	Senin	08.00-09.40	PRAK. PBK (A) (6)	1	PBK1	Julia Pratiwi (6) (B)	Pemrograman 2
3	Senin	08.00-09.40	PRAK. KCB (B) (6)	1	KCB4	Nur Afni Hasanah (6) (C)	Database
4	Senin	09.40-11.20	PRAK. AP (A) (2)	1	AP1	Afifah Rianti (4) (B)	Pemrograman 1
5	Senin	09.40-11.20	PRAK. PW1 (C) (2)	1	PW1 (5)	Dwi Fimoza (6) (A)	Multimedia

Gambar 14. Halaman Setelah *Generate* Jadwal

Pada gambar 13 terlihat terjadi perubahan jadwal, namun perbedaannya dapat dilihat di gambar 4.10. Karena jumlah data yang terlalu banyak yaitu sebanyak 80 data, maka peneliti hanya menampilkan sebagian saja data untuk melihat perubahan dari jadwal yang terjadi. Disini, waktu eksekusi yang sekitar 7 detik pada saat melakukan *generate* jadwal adalah relatif cepat untuk penyusunan suatu jadwal yang cukup kompleks. Kemudian waktu eksekusi bergantung dari proses komputasi saat itu dan juga sistem operasinya.

3) Tahap *Download* Jadwal

Kemudian setelah melakukan proses *generate*, *user* menyimpan hasil *generate* nya kemudian mengolahnya sehingga menyesuaikan dengan jadwal yang akan digunakan. *User* bisa *download* hasil *generate* dengan menekan tombol bertuliskan *Export*. Dan setelah itu *browser* secara otomatis *download* file tersebut serta menyimpannya pada direktori bawaan dari *browser*. Gambar 15 adalah gambar dari hasil *download* jadwal

1	JAM	SKS	MATAKULIAH	SEMESTER	KELAS	KETUA KELAS	RUANG
2	08.00-09.40	1	PRAK. SI (C)	4	R3	Mutiah Khairani	Jaringan 1
3	08.00-09.40	1	PRAK. PBK (A)	6	R1	Julia Pratiwi	Pemrograman 2
4	08.00-09.40	1	PRAK. KCB (B)	6	R2	Nur Afni Hasanah	Database
5	09.40-11.20	1	PRAK. AP (A)	2	R1	Afifah Rianti	Pemrograman 1
6	09.40-11.20	1	PRAK. PW1 (C)	2	R3	Dwi Fimoza	Multimedia
7	09.40-11.20	1	PRAK. SD (A)	2	R1	Andrian	Pemrograman 4
8	09.40-11.20	1	PRAK. PV (A)	4	R1	Dwi Fimoza	Jaringan 1
9	11.20-13.00	1	PRAK. TEKKOM (B)	6	R2	Khairunnada	Jaringan 1
10	13.00-14.40	1	PRAK. PDS (C)	2	R3	Ika Nurul Faradillah	Pemrograman 2
11	13.50-15.30	1	PRAK. PDS (B)	2	R2	Galuh Atika Nabila	Pemrograman 1
12	13.50-15.30	1	PRAK. KDJK (C)	4	R3	Arif Iskandar	Database
13	13.50-15.30	1	PRAK. KGV (C)	4	R3	Dwi Fimoza	Jaringan 1
14	13.50-15.30	1	PRAK. SI (A)	4	R1	Ika Nurul Faradillah	Pemrograman 4
15	13.50-15.30	1	PRAK. KCB (B)	6	R2	Khairunnada	Pemrograman 2

Gambar 15. Hasil *Download* Jadwal

3.7 Analisis Kerja Sistem

Di tahapan ini, peneliti mengerjakan analisis kerja dari sistem yang dilaksanakan di program. Peneliti mengambil beberapa dari sampel data supaya bisa dilihat cara kerja dari algoritma genetika dalam melakukan *generate* jadwal. Beberapa tahapan untuk *generate* jadwal, adalah:

1) Tahap Pembangkitan Jadwal.

Disini sistem mengambil seluruh data yang ada yaitu mata kuliah, hari, jam, pengampu, kelas dan ruang. Dan data yang dipakai adalah data pada tahun ajaran 2018-2019 semester genap. Disini peneliti memakai parameter algoritma genetika sebagai berikut:

Jumlah populasi = 5.

Probabilitas *crossover* = 0.70.

Probabilitas mutasi = 0.40.

Jumlah Generasi = 10000.

Untuk pembentukan dari kromosom awal dilakukan dengan cara acak sebanyak jumlah dari populasi parameter algoritma genetika. Kemudian setelah melakukan pengacakan maka diambil 5 sampel dari tiap kromosom lalu didapatkanlah jadwal yaitu sebagai berikut:

Kromosom 0:

[Dwi Sidik Permana (AP1):Senin:08.00-09.40:Pemrograman 1]

[Rino Subekti (PBK5):Rabu:09.40-11.20:Pemrograman 3]

[Boy Firmansyah (PV6):Kamis :13.00-14.4 :Pemrograman 4]

[Ari Kurniawan (KDJK6):Selasa:08.00-09.40:Database]

[Dwi Sidik Permana (PDS2):Senin:08.00-09.40:Pemrograman 2]

Kromosom 1:

[Boy Firmansyah (PV6):Rabu:13.00-14.40:Jaringan 1]

[Ari Kurniawan (KDJK6): Senin:13.00-14.40:Pemrograman 4]

[Dwi Sidik Permana (PDS2):Kamis:08.00-09.40:Pemrograman 1]

[Dwi Sidik Permana (AP1):Selas :09.40-11.20:Jaringan 2]

[Rino Subekti (PBK5):Jumat:09.40-11.20:Database]

Kromosom 2:

[Rino Subekti (PBK5):Jumat:13.50-15.30:Pemrograman2]

[Dwi Sidik Permana (AP1):Selasa: 3.00-14.40:Database]

[Dwi Sidik Permana (PDS2):Kamis:09.40-11.20:Disain Web]

[Boy Firmansyah (PV6): Senin:13.00-14.40:Jaringan 1]

[Ari Kurniawan (KDJK6):Rabu:08.00-09.40:Jaringan 2]

Kromosom 3:

[Dwi Sidik Permana (AP1):Selasa:09.40-11.20:Pemrograman 3]

[Ari Kurniawan (KDJK6):Senin:08.00-09.40:Jaringan 2]

[Dwi Sidik Permana (PDS2):Kamis:13.00-14.40:Pemrograman 4]

[Rino Subekti (PBK5):Jumat:08.00- 09.40:Disain Web]

[Boy Firmansyah (PV6):Kamis:08.00-09.40:Pemrograman 1]

Kromosom 4:

[Ari Kurniawan (KDJK6):Rabu:08.00-09.40:Jaringan 1]

[Boy Firmansyah (PV6):Selasa : 08.00-09.40:Pemrograman 3]

[Dwi Sidik Permana (AP1):Jumat:09.40-11.20:Pemrograman 1]

[Rino Subekti (PBK5): Kamis:13.00-14.40:Disain Web]

[Dwi Sidik Permana (PDS2):Rabu:08.00-09.40:Pemrograman 4]

2) Tahap Cek *Fitness*

Di tahapan ini setiap dari kromosom akan dihitung nilai *fitness*-nya. Kemudian untuk menentukan nilai dari *fitness* maka menggunakan rumus:

$$Fitness = (1 / 1 + \text{pinalti}).$$

Dimana pinalti dalam program ini yaitu:

- Bentrok pada jadwal mata kuliah wajib praktik dengan jadwal praktikum.
- Bentrok pada jadwal mata kuliah wajib pengampu dengan jadwal mengajarnya.
- Bentrok di hari, jam, serta ruang yang sama antar mata kuliah.
- Bentrok di hari, jam, dan pengampu pada 2 praktikum ataupun lebih. Dan untuk setiap *array* terdapat jumlah maksimum pinalti yaitu sebanyak 4. Jadi tiap kromosom yang memiliki 5 *array* maka memiliki kemungkinan terburuk yaitu pinalti tiap kromosom akan terdapat 20 pinalti.

Pada tiap sampel, maka didapat nilai pinalti dan *fitness*:

- Kromosom 0.
pinalti = 7
 $Fitness = (1/ 1+8) = 0,125$
- Kromosom 1.
pinalti = 13
 $Fitness = (1/ 1+13) = 0,0714$
- Kromosom 2.
pinalti = 11
 $Fitness = (1/ 1+11) = 0,083$
- Kromosom 3.
pinalti = 5
 $Fitness = (1/ 1+5) = 0,33$
- Kromosom 4.
pinalti = 9
 $Fitness = (1/ 1+9) = 0,1$

Dari data-data tersebut, maka dapat disimpulkan yaitu kromosom 3 adalah kromosom terbaik antara 5 kromosom dikarenakan memuat nilai *fitness* yang tertinggi.

3) Tahap Seleksi

Di tahapan ini, akan dipilih kromosom mana yang nanti akan disilangkan terhadap kromosom lain. Sehingga akan terjadi suatu perubahan nilai *array* terhadap suatu komponen. Kemudian Peneliti memakai metode *rank selection* dalam seleksi. Dan untuk melakukan seleksi, yang pertama ditentukan adalah rangking tiap kromosom didasarkan nilai *fitness*, sehingga didapat peringkat:

1. Kromosom 3 = 0.33
2. Kromosom 1 = 0.125
3. Kromosom 4 = 0.1
4. Kromosom 2 = 0.083
5. Kromosom 1 = 0.0714

Karena kromosom 3 meraih peringkat terbaik, sehingga nilai dari peringkat sejumlah populasi, yaitu 5. Serta untuk nilai dari peringkat populasi di bawahnya akan bernilai (n-1). Dan didapatkan nilai peringkatnya:

1. Kromosom 3 = 5
2. Kromosom 0 = 4
3. Kromosom 4 = 3

4. Kromosom 2 = 2
5. Kromosom 1 = 1

Kemudian total nilai *ranking* dari seluruh kromosom ditambahkan, sehingga didapatkanlah semua nilai *ranking* $(5+4+3+2+1) = 15$.

Lalu setelah itu dikerjakan kumulatif dari semua nilai *ranking* yang diperoleh, maka dihasilkan kumulatif:

1. Kromosom 3 = 0-4
2. Kromosom 0 = 5-9
3. Kromosom 4 = 10-12
4. Kromosom 2 = 13-14
5. Kromosom 1 = 15

Kemudian dibuatlah *roulette* bilangan acak bernilai 0-15. *Roulette* dikerjakan sebanyak nilai dari populasi yaitu 5. *Roulette* dilaksanakan untuk melakukan seleksi terhadap kromosom mana yang nanti akan di *crossover*, lalu dipilih pasangan kromosom dari seleksi. Hasil dari *roulette* tersebut yaitu:

1. Kromosom 3 = 3
2. Kromosom 0 = 6
3. Kromosom 4 = 11
4. Kromosom 2 = 4
5. Kromosom 1 = 7

Dan dari Hasil *roulette* tersebut, didapatkan hasil seleksi dari kromosom-kromosom tersebut:

1. [Kromosom 3] X [Kromosom 3]
2. [Kromosom 0] X [Kromosom 0]
3. [Kromosom 4] X [Kromosom 4]
4. [Kromosom 2] X [Kromosom 3]
5. [Kromosom 1] X [Kromosom 0]

Sehingga kromosom-kromosom yang akan melalui tahap proses *crossover* yaitu kromosom 2 dan kromosom 1.

4) Tahap *Crossover*

Di tahapan ini, dimana kromosom yang telah terpilih dari seleksi akan dilakukan proses *crossover*. Sehingga nilai tiap *array* kromosom 2 akan diganti dengan nilai dari kromosom 3. Begitu pun nilai *array* kromosom 1 akan diganti dengan nilai dari *array* kromosom 0. Disini metode yang dipakai oleh peneliti yaitu 2 *cut point*. Dimana artinya diharapkan setidaknya akan ada 2 nilai yang akan berubah dari suatu *array*. Lalu dipilih secara acak 2 nilai *array* yang berubah. Pada penelitian ini, *crossover* dilaksanakan dengan membangkitkan bilangan persentase dari 0% sampai 100%. Disini jumlah *roulette* adalah sebanyak jumlah dari populasi, sehingga *roulette* akan berjumlah 5. Peneliti memakai probabilitas *crossover* 0.70 yang mana dipersentasekan bernilai 70%. Sehingga, jika hasil *roulette* < 70%, maka kemudian nilai dari suatu *array* kromosom tersebut akan mengalami *crossover*.

Untuk proses persilangan kromosom 2:

- Kromosom 2
- [0] hasil *roulette* = 43 %
 - [1] hasil *roulette* = 53 %
 - [2] hasil *roulette* = 69 %
 - [3] hasil *roulette* = 80 %
 - [4] hasil *roulette* = 72 %

Maka nilai *array* 0,1,2 pada kromosom 2 disilangkan dengan nilai *array* dari kromosom 3 dengan 2 *point cut* dengan cara acak.

[Kromosom 2] X [Kromosom 3].

Sehingga hasil dari kromosom baru adalah:

Kromosom 2:

[Rino Subekti (PBK5):Selasa:13.50-15.30: Pemrograman 3]

[Dwi Sidik Permana (AP1): Senin:13.00-14.40: Jaringan 2]

[Dwi Sidik Permana (PDS2):Kamis :13.00-14.40 :Pemrograman 4]

[Boy Firmansyah (PV6):Senin:13.00-14.40 : Jaringan 1]

[Ari Kurniawan (KDJK6):Rabu:08.00-09.40:Jaringan 2]

Untuk proses persilangan dari kromosom 1:

Kromosom 1 :

[0] hasil *roulette* = 73 %

[1] hasil *roulette* = 33 %

[2] hasil *roulette* = 19 %

[3] hasil *roulette* = 99 %

[4] hasil *roulette* = 89 %

Maka nilai dari *array* 1,2 pada kromosom 1 disilangkan dengan nilai *array* dari kromosom 0 dengan 2 *point cut* dengan cara acak.

[Kromosom 1] X [Kromosom 0].

Sehingga hasil dari kromosom baru adalah:

Kromosom 1:

[Boy Firmansyah (PV6): Senin:08.00-09.40:Jaringan 1]

[Ari Kurniawan (KDJK6):Senin:09.40-11.20: Pemrograman3]

[Dwi Sidik Permana (PDS2):Kamis:08.00-09.40:Pemrograman 1]

[Dwi Sidik Permana (AP1):Selasa : 09.40-11.20:Jaringan 2]

[Rino Subekti (PBK5):Jumat:09.40-11.2 :Database]

5) Tahap Mutasi

Di tahap ini, nilai *array* suatu kromosom diganti dengan nilai lain yang merupakan nilai barunya. Disini pemilihan *array* akan dipilih secara acak. Begitu pun nilai barunya akan didapat secara acak. Kegiatan pertama adalah membangkitkan nilai persentase acak dari 0% - 100% sebagai *roulette*. *Roulette* dikerjakan sebanyak jumlah populasi, yaitu 5. Peneliti memakai probabilitas mutasi yaitu 40%. Artinya jika pada nilai persentase acak < 40%, maka disimpulkan kromosom terjadi mutasi. Jika pada suatu kromosom terjadi mutasi, maka setelahnya akan dibangkitkan bilangan acak dari 0 sampai jumlah *array* - 1. Karena jumlah *array* = 5, maka angka *random* yang dibangkitkan bernilai 0 - 4. Lalu dilakukan *roulette* terhadap bilangan 0 - 4. Hasil *roulette* yaitu nilai *array* yang akan berubah. Hasil *roulette* dari tiap kromosom yaitu:

Kromosom 0 : hasil *roulette* = 83 %

Kromosom 1 : hasil *roulette* = 36 %

Kromosom 2 : hasil *roulette* = 64 %

Kromosom 3 : hasil *roulette* = 76 %

Kromosom 4 : hasil *roulette* = 92 %

Dari data tersebut, maka hanya kromosom 1 yang mengalami mutasi, dimana kromosom lainnya tidak terjadi mutasi. Oleh karenanya setelah itu dibangkitkan bilangan acak dari 0-4 untuk memilih *array*

yang terjadi mutasi. Kemudian hasil dari *roulette* yaitu 3. Sehingga kromosom 1 *array* 3 akan mengalami mutasi. Nilai *array* 3 didapatkan dari pengacakan terhadap data hari, jam, dan ruang yang terdapat dalam *database*. Nilai kromosom yang baru yaitu:

Kromosom 1:

[Boy Firmansyah (PV6):Senin :08.00-09.40 : Jaringan 1]
[Ari Kurniawan(KDJK6):Senin:09.40-11.20:Pemrograman3]
[Dwi Sidik Permana (PDS2):Kamis:08.00-09.40:Pemrograman 1]
[Dwi Sidik Permana (AP1):Selasa:09.40-11.20:Jaringan 2]
[Rino Subekti (PBK5): Jumat : 09.40-11.20:Database]

Maka hasil kromosom yang baru yaitu :

Kromosom 0:

[Dwi Sidik Permana (AP1):Senin:08.00-09.40:Pemrograman 1]
[Rino Subekti (PBK5):Rabu:09.40-11.20:Pemrograman 3]
[Boy Firmansyah (PV6):Kamis:13.00-14.40:Pemrograman 4]
[Arif Iskandar (KDJK6):Selasa:08.00-09.40:Database]

Kromosom 1:

[Boy Firmansyah (PV6): Senin:08.00-09.40:Jaringan 1]
[Arif Iskandar(KDJK6):Senin:09.40-11.20:Pemrograman3]
[Dwi Sidik Permana(PDS2):Kamis:08.00-09.40:Pemrograman 1]
[Dwi Sidik Permana(AP1):Selasa:09.40-11.20:Jaringan 2]
[Rino Subekti (PBK5):Jumat:09.40-11.20:Database]

Kromosom 2:

[Rino Subekti (PBK5): Selasa : 13.50-15.30:Pemrograman 3]
[Dwi Sidik Permana (AP1):Senin:13.00-14.40:Jaringan 2]
[Dwi Sidik Permana (PDS2):Kamis:13.00-14.40:Pemrograman 4]
[Boy Firmansyah (PV6):Senin:13.00-14.40:Jaringan 1]
[Ari Kurniawan (KDJK6):Rabu:08.00-09.40:Jaringan 2]

Kromosom 3 :

[Dwi Sidik Permana(AP1):Selasa:09.40-11.20:Pemrograman 3]
[Ari Kurniawan (KDJK6):Senin:08.00-09.40:Jaringan 2]
[Dwi Sidik Permana (PDS2):Kamis:13.00-14.40:Pemrograman 4]
[Rino Subekti (PBK5):Jumat:08.00- 09.40:Disain Web]
[Boy Firmansyah (PV6): Kamis:08.00-09.40:Pemrograman 1]

Kromosom 4 :

[Ari Kurniawan (KDJK6):Rabu:08.00-09.40:Jaringan 1]
[Boy Firmansyah (PV6):Selasa:08.00-09.40:Pemrograman 3]
[Dwi Sidik Permana (AP1):Jumat:09.40-11.20:Pemrograman 1]
[Rino Subekti (PBK5):Kamis:13.00-14.40:Disain Web]
[Dwi Sidik Permana (PDS2):Rabu:08.00-09.40:Pemrograman 4]

Kemudian dilakukan perhitungan kembali nilai *fitness* yaitu:

- Kromosom 0.
 pinalti = 7
 Fitness = $(1/ 1+8) = 0,125$
- Kromosom 1.
 pinalti = 10

$$Fitness = (1/ 1+10) = 0,091$$

- Kromosom 2.
pinalti = 8
 $Fitness = (1/ 1+8) = 0,111$
- Kromosom 3.
pinalti = 5
 $Fitness = (1/ 1+5) = 0,33$
- Kromosom 4.
pinalti = 9
 $Fitness = (1/ 1+9) = 0,1$

Selesailah proses untuk generasi pertama. Dikarenakan nilai *fitness* kromosom 3 adalah nilai *fitness* tertinggi, sehingga kromosom 3 adalah jadwal yang dianggap mendekati optimum. Prosesnya akan diulang terus sampai mencapai generasi yang diharapkan.

Disini peneliti memakai probabilitas *crossover* = 0.70, probabilitas mutasi = 0.40 dan jumlah generasi = 10.000 sehingga didapatkan jadwal yang cukup optimal seperti terlihat pada gambar 15 pada pengujian.

3.8 Perbandingan *Input-an* Nilai Parameter Algoritma Genetika

Disini, peneliti mencoba membandingkan suatu pengaruh *input-an* parameter dalam sistem dengan nilai hasil dari *generate* jadwal. Dimana pertama peneliti membandingkan nilai *input-an* parameter dari jumlah generasi lalu menjadikan waktu eksekusinya sebagai acuan yaitu apakah *input-an* tersebut baik ataukah tidak. Nilai populasi dari probabilitas *crossover* serta probabilitas mutasi dibuat sama, yaitu bernilai 10 untuk nilai populasi, bernilai 0.70 untuk probabilitas *crossover*, dan bernilai 0.40 untuk probabilitas mutasi. Berikut ini nilai jumlah generasi yang akan dibandingkan:

1. Jumlah generasi = 3000.
2. Jumlah generasi = 5000.
3. Jumlah generasi = 8000.
4. Jumlah generasi = 10000.

Berikut adalah gambar dari hasil *generate* jumlah generasi:

Penjadwalan

Beranda / Penjadwalan

Proses Export to Excel

Jumlah Populasi
10

Probabilitas CrossOver
0.70

Probabilitas Mutasi
0.40

Jumlah Generasi
3000

Waktu Eksekusi 6.1966419219971

Nilai Fitness Terbaik 0.5

Gambar 16. Hasil Untuk 3000 Generasi

Penjadwalan

Beranda / Penjadwalan

Proses Export to Excel

Jumlah Populasi
10

Probabilitas CrossOver
0.70

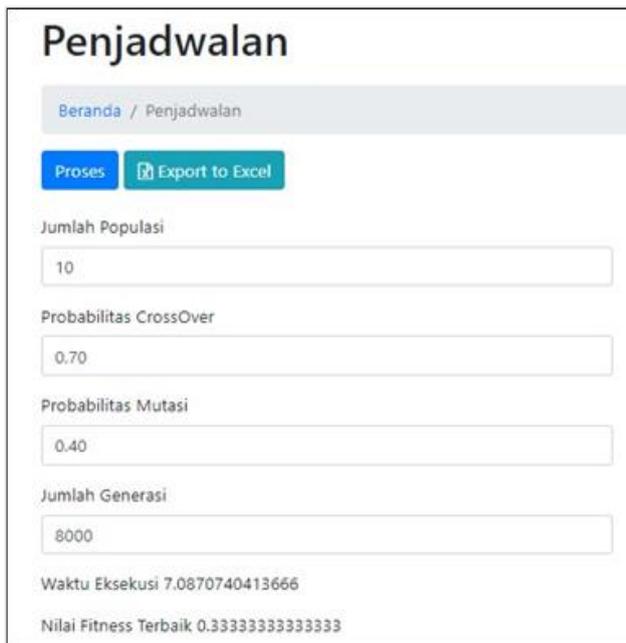
Probabilitas Mutasi
0.40

Jumlah Generasi
5000

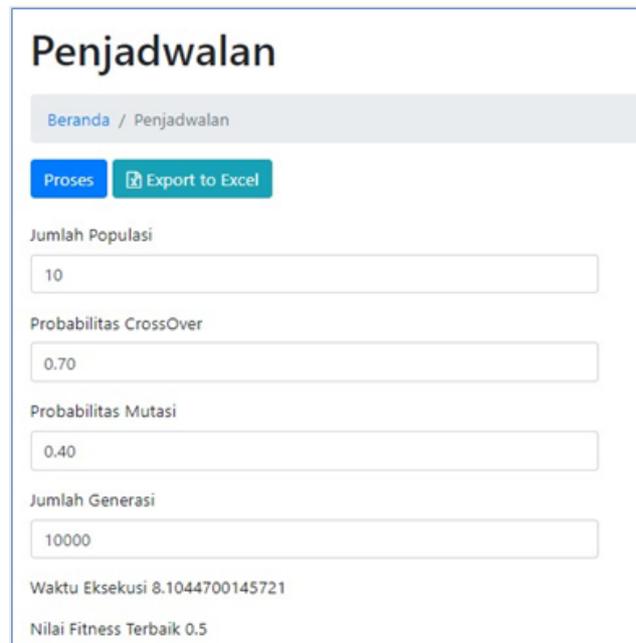
Waktu Eksekusi 9.5773341655731

Nilai Fitness Terbaik 0.5

Gambar 17. Hasil Untuk 5000 Generasi



Gambar 18. Hasil Untuk 8000 Generasi



Gambar 19 Hasil Untuk 10000 Generasi

Hasil dari percobaan dapat dilihat pada tabel berikut:

Tabel 3. Perbandingan Jumlah Generasi

No	Jumlah Populasi	Probabilitas Crossover	Probabilitas Mutasi	Jumlah Generasi	Waktu Eksekusi	Nilai Fitness
1	10	0.70	0.40	3000	6.19664	0.5
2	10	0.70	0.40	5000	9.57733	0.5
3	10	0.70	0.40	8000	7.08707	0.333
4	10	0.70	0.40	10000	8.10447	0.5

Berdasarkan dari percobaan di atas, maka dapat disimpulkan jumlah generasi 3000 telah menghasilkan jadwal cukup baik. Dikarenakan jumlah generasinya yang lebih sedikit, maka proses *generate* menjadi lebih cepat. Dan tentunya karena sistem *generate* dari jadwal yang bersifat heuristik, sehingga nilai jadwal dapat berubah tergantung dari pengacakan yang terjadi. Peneliti mencoba membandingkan nilai parameter dari probabilitas *crossover* serta probabilitas mutasi untuk dapat melihat *input-an* nilai parameter mana lebih baik. Kali ini, peneliti menetapkan *input-an* nilai jumlah dari populasi sebanyak 10 serta jumlah populasi adalah sebanyak 3000. Kemudian untuk nilai probabilitas *crossover* serta probabilitas mutasi yaitu sebagai berikut:

- 1) Probabilitas *crossover* = 0.50 dan probabilitas mutasi = 0.20.
- 2) Probabilitas *crossover* = 0.60 dan probabilitas mutasi = 0.30.
- 3) Probabilitas *crossover* = 0.70 dan probabilitas mutasi = 0.40.
- 4) Probabilitas *crossover* = 0.80 dan probabilitas mutasi = 0.50.

Berikut ini adalah gambar hasil nilai probabilitas *crossover* serta probabilitas mutasi:

Penjadwalan

Beranda / Penjadwalan

Proses [Export to Excel](#)

Jumlah Populasi
10

Probabilitas CrossOver
0.50

Probabilitas Mutasi
0.30

Jumlah Generasi
3000

Waktu Eksekusi 7.0096321105957

Nilai Fitness Terbaik 0.25

Gambar 20. Hasil *Crossover* Dan Mutasi 1

Penjadwalan

Beranda / Penjadwalan

Proses [Export to Excel](#)

Jumlah Populasi
10

Probabilitas CrossOver
0.60

Probabilitas Mutasi
0.30

Jumlah Generasi
3000

Waktu Eksekusi 8.4341638088226

Nilai Fitness Terbaik 0.5

Gambar 21. Hasil *Crossover* Dan Mutasi 2

Penjadwalan

Beranda / Penjadwalan

Proses [Export to Excel](#)

Jumlah Populasi
10

Probabilitas CrossOver
0.70

Probabilitas Mutasi
0.40

Jumlah Generasi
3000

Waktu Eksekusi 5.4773619174957

Nilai Fitness Terbaik 0.5

Gambar 22. Hasil *Crossover* Dan Mutasi 3

Penjadwalan

Beranda / Penjadwalan

Proses [Export to Excel](#)

Jumlah Populasi
10

Probabilitas CrossOver
0.80

Probabilitas Mutasi
0.50

Jumlah Generasi
3000

Waktu Eksekusi 9.5633339881897

Nilai Fitness Terbaik 0.25

Gambar 23. Hasil *Crossover* Dan Mutasi 4

Hasil dari percobaan dapat dilihat pada tabel 4.

Tabel 4. Perbandingan *Crossover* Dan Mutasi

No	Jumlah Populasi	Probabilitas <i>Crossover</i>	Probabilitas Mutasi	Jumlah Generasi	Waktu Eksekusi	Nilai <i>Fitness</i>
1	10	0.50	0.20	3000	7.00096	0.25
2	10	0.60	0.30	3000	8.43416	0.5
3	10	0.70	0.40	3000	5.47736	0.5
4	10	0.80	0.50	3000	9.56333	0.25

Dari percobaan yang dilakukan bisa disimpulkan yaitu nilai probabilitas *crossover* = 0.70 serta probabilitas mutasi = 0.40 akan menghasilkan jadwal terbaik sebab memiliki nilai *fitness* yang terbaik untuk jumlah generasi yang bernilai 3000.

Meskipun percobaan kedua dan ketiga memiliki nilai *fitness* yang sama untuk nilai probabilitas *crossover* = 0.60 serta probabilitas mutasi = 0.40, akan tetapi proses waktu eksekusinya lebih cepat.

Sehingga, dari 2 percobaan yang ada, maka *input-an* parameter jumlah generasi = 3000, lalu probabilitas *crossover* = 0.70 serta probabilitas mutasi = 0.40 dapat disimpulkan menghasilkan jadwal yang cukup optimum.

IV. PENUTUP

Dari uraian dan pembahasan yang dilaksanakan berdasarkan bab-bab sebelumnya, untuk itu dapat disimpulkan dari sistem penjadwalan mata kuliah dengan menggunakan algoritma genetika ini adalah sebagai berikut:

- 1) Dengan menggunakan algoritma genetika di dalam penjadwalan mata kuliah akan dapat menghasilkan jadwal yang cukup optimum.
- 2) *Input* parameter dari algoritma genetika terbukti sangat berpengaruh sekali terhadap baik atau tidaknya nilai jadwal yang di *generate*.
- 3) *Input* parameter dari algoritma genetika dengan jumlah generasi sebesar 3000, dan probabilitas *crossover* sebesar 0.60 serta probabilitas mutasi sebesar 0.30 dapat menghasilkan jadwal terbaik didasarkan atas percobaan yang dibuat oleh peneliti.
- 4) Proses dari *generate* jadwal akan relatif cepat dikarenakan menggunakan waktu komputasi yang sedikit serta lebih mengandalkan suatu probabilitas.
- 5) Di dalam penginputan dari parameter algoritma genetika akan memiliki pengaruh yang besar dengan nilai hasil *generate* jadwal, dimana bisa saja jadwal akan menjadi lebih baik ataupun mungkin menjadi lebih buruk.
- 6) Dengan menggunakan algoritma genetika serta sistem yang dibuat oleh peneliti, maka seorang *user* akan dapat memiliki banyak pilihan di dalam memilih jadwal yang nantinya akan digunakan dikarenakan setiap *generate* akan menghasilkan jadwal yang benar-benar berbeda.

REFERENSI

- Ashari, Imam Ahmad, 2016, "Perbandingan performansi algoritma genetika dan algoritma ant colony optimization dalam optimasi penjadwalan", Diss. Universitas Negeri Semarang
- Babaei, H., Karimpour, J., & Hadidi, A., 2015, "A survey of approaches for university course timetabling problem", Computers & Industrial Engineering, pp. 8.
- Deep, Kusum, et al, 2009, "A real coded genetic algorithm for solving integer and mixed integer optimization problems." Applied Mathematics and Computation Vol 212 No.2, pp. 505-518.
- Elsayed, Saber M., Ruhul A. Sarker, and Daryl L. Essam, 2014, "A new genetic algorithm for solving optimization problems." Engineering Applications of Artificial Intelligence Vol 2 No 7, pp. 57-69.
- Hamad, Safwat A., and Fatma A. Omara, 2016, "Genetic-based task scheduling algorithm in cloud computing environment." International Journal of Advanced computer Science and Applications Vol 7 No.4, pp. 550-556.
- Hermawanto, Denny, 2014, "Algoritma Genetika dan contoh aplikasinya." Publikasi Lembaga Ilmu Pengetahuan Indonesia, pp. LIPI-7003
- Jain, Ashish, Suresh Jain, and P. K. Chande, 2010, "Formulation of genetic algorithm to generate good quality course timetable." International Journal of Innovation, Management and Technology Vol 1 No.3, pp. 248.

- Janata, Ari, and Elin Haerani, 2015, "Sistem Penjadwalan Outsourcing Menggunakan Algoritma Genetika (Studi Kasus: PT. Syarikatama)." *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi* Vol 1 No.1, pp. 17-24.
- Kumar, Manoj, et al, 2010, "Genetic algorithm: Review and application." *International Journal of Information Technology and Knowledge Management* Vol 2 No.2, pp. 451-454.
- Puspaningrum, Wiga Ayu, Arif Djunaidy, and Retno Aulia Vinarti, 2013, "Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS." *Jurnal Teknik ITS* Vol 2 No.1, pp. A127-A131.
- Rahman, F. W, 2011, "Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Kuliah (Studi Kasus di Program Studi Teknik Industri Undip)", Diss. Diponegoro University.
- Salhi,Saïd, 2017, "Heuristic Search The Emerging Science of Problem Solving", Canterbury, United Kingdom: Springer Nature.
- Sembiring,Martinelly, 2013, "Penjadwalan Perkuliahan Dengan Menggunakan Algoritma Genetika (Studi Kasus: S-1 Ilmu Komputer Universitas Sumatera Utara)", *123dok.com* Vol 6 no.1, pp. 1-10.
- Widodo, Agus Wahyu, and Wayan Firdaus Mahmudy, 2010, "Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner.", *Jurnal Genetika (GA)* Vol 3 No.4, pp. 5.