

Rancang Bangun Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetika (Studi Kasus : Universitas CIC)

Syifa Ulkarim¹, Marsani Asfi², Tiara Eka Putri³

Universitas Catur Insan Cendekia

Kota Cirebon, Indonesia

¹syifaulkarim@gmail.com, ²marsani.asfi@gmail.com, ³tiara.eka@cic.ci.id

Diterima : 30 Agustus 2020

Disetujui : 28 September 2020

Abstrak — Universitas Catur Insan Cendekia (UCIC) adalah perguruan tinggi yang berlokasi di Jl. Kesambi No. 202 Kota Cirebon dan berada pada naungan yayasan Catur Insan Cendekia. Dalam upaya melakukan penjadwalan kuliah Universitas Catur Insan Cendekia masih dibuat secara semi manual sehingga terbilang masih kurang efisien, dan sering kalinya terjadi bentrok antar jadwal kuliah sehingga dilakukannya revisi berkali-kali untuk mencapai jadwal kuliah yang sesuai. Untuk mengatasi permasalahan tersebut diperlukan suatu sistem yang dapat melakukan optimasi jadwal kuliah yang lebih baik. Metode algoritma genetika adalah metode yang diterapkan dalam melakukan optimasi penjadwalan kuliah. Metode perancangan program yang digunakan adalah perancangan dengan pendekatan berorientasi objek yaitu menggunakan *Unified Modelling Language*. Bahasa pemrograman yang digunakan adalah *Hypertext Preprocessor*, *MySQL*, dan *laravel* sebagai kerangka kerja dari rancangan pembuatan program yang akan dibangun. Hasil dari penelitian ini adalah sistem yang mampu menghasilkan penjadwalan kuliah, sistem ini dapat melakukan optimasi penjadwalan dengan hasil yang lebih efisien sehingga dapat memudahkan tugas BAAK Universitas CIC.

Kata Kunci : Algoritma Genetika, Optimasi, Penjadwalan kuliah, Universitas CIC.

I. PENDAHULUAN

Universitas Catur Insan Cendekia (UCIC) adalah perguruan tinggi yang berlokasi di Jl. Kesambi No. 202 Kota Cirebon dan berada pada naungan yayasan Catur Insan Cendekia (CIC). Universitas Catur Insan Cendekia memiliki 2 Fakultas yaitu Fakultas Teknologi Informasi dan Fakultas Ekonomi Bisnis. Dari 2 fakultas tersebut tersebar 8 program studi yang terdiri dari program sarjana dan diploma.

Sistem penjadwalan perkuliahan di Universitas Catur Insan Cendekia masih dibuat secara semi manual sehingga perlu adanya sistem penjadwalan otomatis agar tidak terjadinya bentrok jadwal perkuliahan dengan jadwal perkuliahan lainnya. Sistem pembuatan jadwal perkuliahan di Universitas Catur Insan Cendekia saat ini belum bisa membagi jadwal dengan otomatis sesuai dengan aturan yang di tentukan. Seringkali adanya bentrok antar kelas, sehingga jadwal direvisi berkali-kali. Dalam proses

manual kesalahan lainnya adalah terjadinya *human error* dimana BAAK dalam menyusun jadwal sering memasukkan data yang berulang - ulang sehingga memperlambat proses pembuatan jadwal perkuliahan.

Oleh karena itu dibutuhkan sistem berbasis komputer untuk melakukan proses penjadwalan kuliah yang cepat dan akurat dengan menerapkan algoritma yang dapat menyusun jadwal kuliah secara otomatis. Salah satu algoritma yang dapat dimanfaatkan dalam pembuatan sistem penjadwalan kuliah adalah algoritma genetika.

Beberapa penelitian penjadwalan dengan algoritma genetika adalah penelitian [1] dan [2]. Pada penelitian tersebut dapat disimpulkan bahwa solusi masalah penjadwalan yang kurang efisien dapat diselesaikan dengan metode algoritma genetika yaitu melalui proses kawin silang (*cross over*), mutasi, inversi, dan proses lainnya pada algoritma genetika [1] [2].

Berdasarkan permasalahan dan penelitian terdahulu yang telah dilakukan, maka tujuan penelitian ini dilakukan adalah membangun sistem penjadwalan kuliah pada Universitas CIC menggunakan algoritma genetika. Algoritma genetika dipilih karena sudah banyak dipakai dalam penelitian terkait dan memiliki akurasi yang tinggi dalam optimasi pembuatan penjadwalan.

II. LANDASAN TEORI

Algoritma genetika adalah salah satu metode dalam perkembangan dunia IT dalam bidang kecerdasan buatan. Salah satu fungsi metode ini adalah untuk optimasi penjadwalan [3]. Terdapat beberapa komponen dalam algoritma genetika dalam [4] yaitu:

a. Skema pengkodean (*encoding*)

Secara umum algoritma genetika menggunakan jenis bilangan real, desimal atau biner, yang biasa disebut *real-number encoding*. Contoh penerapan bilangan *real-number encoding* pada algoritma genetika ditunjukkan pada Gambar 1.

Kromosom A	2 1 3 5 6 8 7 4 9
Kromosom B	6 1 3 4 2 9 5 7 8

Gambar 1. Representasi *real-number encoding*

b. Nilai *fitness*

Dalam algoritma genetika dapat disebutkan bahwa kemampuan suatu individu dalam bertahan dapat diukur berdasarkan nilai *fitness*-nya. Semakin tinggi nilai *fitness* dalam suatu individu, maka akan semakin tinggi peluang individu tersebut akan bertahan hidup. Nilai *fitness* suatu individu dapat dihitung menggunakan rumus *fitness* dalam [5] yaitu sebagai berikut :

$$fitness = \frac{1}{1 + (\sum BD + \sum BK + \sum BR)} \quad (1)$$

Keterangan :

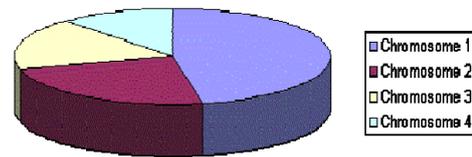
BD = Banyaknya bentrok antara jumlah dosen dan mata kuliah

BK = Banyaknya bentrok kelas perkuliahan yang digunakan

BR = Banyaknya bentrok jumlah ruang yang digunakan

c. Seleksi orang tua

Pada metode seleksi orang tua ini, setiap orang tua dipilih yaitu berdasarkan nilai *fitness* yang terbesar. Contohnya yaitu peluang terpilihnya suatu individu pada roda *roulette* yang diilustrasikan pada Gambar 2.



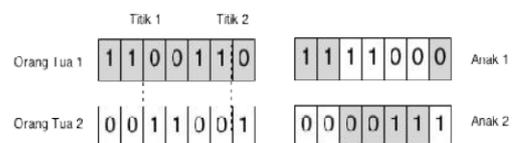
Gambar 2. Seleksi roda *roulette*

d. Seleksi roda *roulette*

Seleksi ini adalah seleksi yang merupakan bagian dari seleksi rangking. Namun ada yang membedakannya yaitu pada seleksi roda *roulette* adalah pemilihan individu yang akan digunakan untuk bereproduksi tidak berdasarkan pada besarnya nilai *fitness* dari suatu populasi.

e. *Crossover*

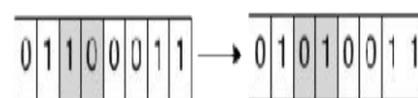
Pada proses *crossover* akan menghasilkan berbagai variasi gen. Dimana terdapat dua individu dipilih sebagai induk. Setelah didapatkan dua individu induk, maka selanjutnya adalah menentukan pusat yang akan dilakukan proses *crossover*. Pada proses *crossover* ini akan menggunakan banyak sumber titik untuk melakukan persilangannya. Contoh *crossover* banyak titik ditunjukkan pada Gambar 3.



Gambar 3. Contoh Pindah Silang Banyak Titik

f. Mutasi

Mutasi adalah proses yang dilakukan untuk mengembalikan nilai bit yang hilang akibat proses *crossover*. Mutasi diterapkan dengan nilai peluang yang sangat kecil. Proses mutasi pada algoritma genetika dapat dilihat pada Gambar 4.



Gambar 4. Contoh mutasi tingkat gen

g. Elitisme

Proses elitisme adalah proses membuat *copy* atau suatu salinan dalam algoritma genetika yang tujuannya adalah untuk mempertahankan agar setiap individu yang memiliki nilai fitness yang tinggi tidak akan hilang pada saat proses evolusi berlangsung.

- h. Penggantian populasi
 Pada proses algoritma genetika, maka setiap N individu pada suatu generasi dapat digantikan sekaligus oleh N individu baru *crossover* ataupun melalui proses mutasi. Proses inilah yang disebut sebagai penggantian populasi.

III. ANALISIS DAN PERANCANGAN

1. Analisis Data

Pada perancangan proses penjadwalan kuliah ini akan diimplementasikan algoritma genetika yang terdiri dari beberapa tahapan metode seperti pembangkitan kromosom, pembangkitan populasi, perancangan nilai *fitness* individu, seleksi, *crossover*, mutasi, dan *update* generasi.

Tabel 1. Data Mata Kuliah

Kode	Mata Kuliah	SKS	Semester
MO1	PBO	4	5
MO2	Jaringan Komputer	3	4
MO3	Algoritma	4	2
MO4	Keamanan Komputer	3	6
MO5	T Multimedia	2	6
MO6	Kalkulus	3	2

Dalam pembangkitan kelas ini penulis ilustrasikan kedalam 10 kelas dalam sehari yaitu hari senin sebanyak 9 (07:15-16:35) sesi waktu yang di sediakan. Berikut langkah-

langkah pembangkitan kelas penjadwalan kuliah algoritma genetika :

Pembangkitan kromosom/pengkodean
 Pembangkitan kromosom dilakukan dengan memanfaatkan aturan umum (*hard constraint*) dan diharapkan tiap kromosom yang dibangkitkan tersebut dapat memenuhi semua aturan yang ada di dalam algoritma genetika. Kromosom merepresentasikan kandidat solusi jadwal yang akan dihasilkan yang terbentuk dari kumpulan kelas, waktu, dan ruang. Ilustrasi waktu yang digunakan pada contoh ini adalah 1 hari yaitu hari Senin dengan 9 sesi (07:15-16:35). Waktu perkuliahan yang berjumlah 9 sesi ditunjukkan pada Tabel 2 di bawah ini yaitu :

Tabel 2. Sesi waktu kuliah.

Waktu	Senin
07.15 – 08.35	1
08.15 – 09.35	2
09.15 – 10.35	3
10.15 – 11.35	4
11.15 – 12.35	5
12.15 – 13.35	6
13.15 – 14.35	7
14.15 – 15.35	8
15.15 – 16.35	9

Dari 9 sesi waktu yang didapatkan kemudian dikombinasikan dengan ruang menghasilkan matriks ruang waktu. Untuk pembangkitan gen dilakukannya pengkodean atau inialisasi pada data dosen pengampu. Data terlihat pada Tabel 3. Di bawah ini :

Tabel 3. Data dosen pengampu

Nama Dosen	Kode Matkul	Jumlah Peserta	Kelas	Gen
Chairun Nas, M.Kom.	M01	30	TI-SE-1/6	1
Ilwan Syafrinal, M.Kom.	M02	20	TI-1/4	2
Ilwan Syafrinal, M.Kom.	M02	25	TI-2/4	3
Ilwan Syafrinal, M.Kom.	M02	22	TI-3/4	4
Ilwan Syafrinal, M.Kom.	M01	31	TI-SCN-1/6	5
Kusnadi, M.Kom.	M03	27	TI-1/2	6
Kusnadi, M.Kom.	M04	30	TI-SE-1/6	7
Kusnadi, M.Kom.	M05	30	TI-SE-1/6	8
...

Dari pembentukan gen di atas maka selanjutnya dilakukannya pembentukan kromosom. Kromosom matriks ruang waktu

dimodelkan seperti yang ditunjukkan pada Tabel 4. Di bawah ini :

Tabel 4. Pembangkitan kromosom

Ruangan	Senin								
	1	2	3	4	5	6	7	8	9
102	1	2	3	4	5	6	7	8	9
103	10								
...

Matriks ruang waktu berisi indeks yang merupakan angka yang berurutan dari sesi pertama hingga terakhir pada hari Senin untuk semua daftar ruang mata kuliah. Tiap indeks pada matriks ruang waktu mewakili sesi, hari,

dan ruang. Kemudian secara acak akan mengisi gen-gen pada kromosom dan masing-masing nilainya berbeda seperti yang ditunjukkan pada Tabel 5. Dibawah ini :

Tabel 5. Pembangkitan kromosom pengacakan gen generasi 1

Ruangan	Senin								
	1	2	3	4	5	6	7	8	9
102	5	2	8	5	4	8	1	8	8
103	8	3	2	3	4	9	10	5	10
104	8	3	10	9	1	9	2	3	7
105	10	6	5	1	4	2	3	10	1
...

Pada saat penentuan indeks matriks ruang waktu pada sebuah gen dapat dilakukan lebih dari satu kali pengacakan. Saat pengacakan, indeks matriks ruang waktu ini dapat gugur ketika indeks tersebut tidak memenuhi salah satu aturan umum (*hard constraints*) penjadwalan. Pada Tabel 10 di

lihat contoh gen yang melanggar aturan umum gen 8,4,3,9 yaitu bentrok waktu, ruangan dan gen. Sehingga dilakukan pengacakan kembali hingga indeks benar-benar layak atau memenuhi semua aturan umum. Model kromosom dapat ditunjukkan seperti pada Tabel 6. dan Tabel 7. Di bawah ini:

Tabel 6. Pembangkitan kromosom pengacakan indeks

Ruangan	Senin								
	1	2	3	4	5	6	7	8	9
...
103	8	3	2	3	4	9	10	5	10
104	8	3	10	9	1	9	2	3	7
...

Tabel 7. Pembangkitan hasil pengacakan indeks

Ruangan	Senin								
	1	2	3	4	5	6	7	8	9
...
103				7					
104						9			
...

Pada tabel di atas di jelaskan perpindahan indeks dari kondisi pada Tabel 6. yang kemudian menjadi gen baru yang terbentuk yang di tunjukkan pada Tabel 7. Di atas.

Pembangkitan populasi

Metode pembangkitan kromosom yang dilakukan pada algoritma genetika adalah metode pengacakan secara random permutasi pada indeks matriks ruang waktu dengan menggunakan pengkodean nilai. Dalam

pencarian gen secara acak dilakukan pengecekan terhadap aturan umum hingga benar-benar memenuhi aturan tersebut. Dari gen-gen yang telah memenuhi aturan-aturan umum, maka akan digabungkan secara terurut menjadi satu kromosom. Slot-slot kromosom mewakili kelas mata kuliah dan gen mewakili ruang dan waktu yang akan dijadwalkan pada kelas tersebut. Pembangkitan populasi ditunjukkan pada Tabel 8. Di bawah ini :

Tabel 8. Pembangkitan *fitness*.

Ruangan	Space	Senin								
		1	2	3	4	5	6	7	8	9
102	36		9 31					1 30		
103	55				7 30					
104	28						9 31			8 30
105	43		6 27	5 31						
106	40					4 22				
107	40									
201	40			3 25						
202	33						10 27			

I. EVALUASI FITNESS

Nilai *fitness* pada individu menggunakan nilai antara 0 dan 1, dengan 0 jika sebuah kromosom melanggar semua aturan umum, dan 1 jika semua aturan umum

terpenuhi oleh kromosom. Masing-masing aturan khusus memiliki nilai sebagai representasi dari masing-masing kromosom yang bernilai antara 0 dan 1. Fungsi *fitness* yang digunakan adalah seperti pada persamaan 1. Maka didapat hasil sebagai berikut :

Tabel 9. Pembangkitan populasi generasi 2

Ruangan	Space	Senin								
		1	2	3	4	5	6	7	8	9
102	36		9 31					1 30		
103	55				7 30					
104	28						9 31			8 30
105	43		6 27	5 31						
106	40					4 22				
107	40									
201	40			3 25						
202	33						10 27			

Rumus : $fitness = \frac{1}{1 + (0+1+1)} = 0,33$

Maka didapatkan :

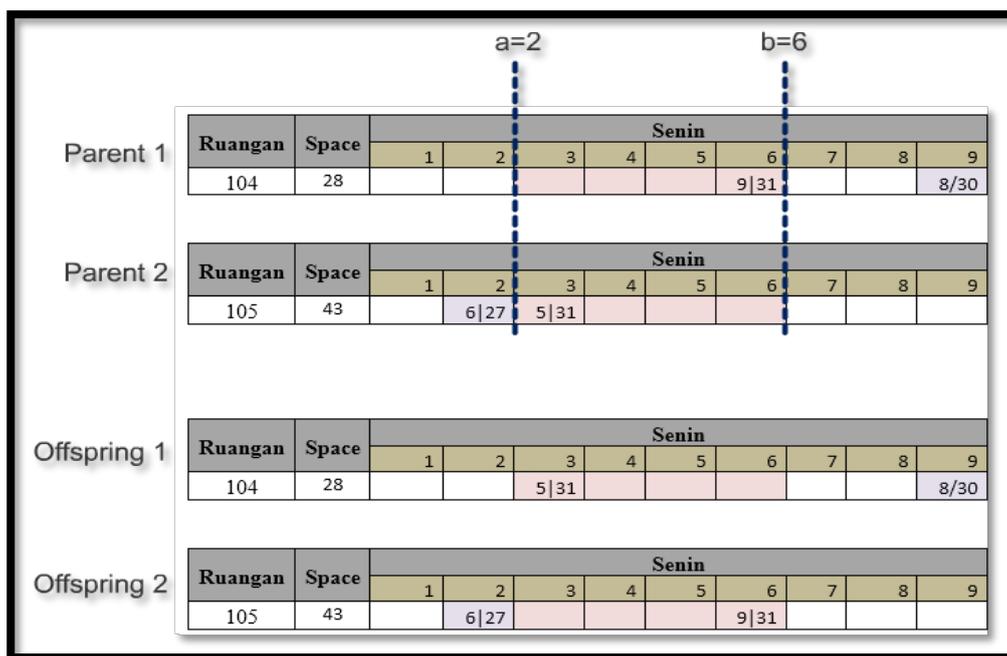
Tabel 10. Nilai *fitness* kromosom

Inisialisasi	Gen	Keterangan	<i>Fitness</i>
BM	9 31	Banyaknya ruangan yang tidak memenuhi	0
BR	<i>Null</i>	Banyaknya bentrok ruang yang digunakan	1
BK	<i>Null</i>	Banyaknya bentrok kelas mata kuliah	1
Total <i>fitness</i>			0,33

Dalam langkah seleksi ini yaitu menyeleksi dari beberapa generasi yang terbentuk. seperti yang di tunjukkan pada langkah 3 yaitu mendapatkan generasi 2 di mana hasil fitnessnya 0,33 yaitu belum memenuhi aturan umum.

Metode perkawinan silang (*crossover*) yang digunakan adalah metode n-point *crossover* dengan 2 titik (*2-point crossover*) yang kemudian dikawinkan silang menggunakan *2-point crossover* berdasarkan titik potong yang telah didapatkan hingga akhirnya didapatkan *offspring*. Apabila terjadi

duplikasi offspring, yakni terdapat gen yang sama dalam satu kromosom atau dalam artian terdapat jadwal ruang dan waktu yang sama, maka salah satu gen tersebut secara random akan memilih slot yang kosong pada matriks ruang waktu yang belum digunakan sehingga kromosom *offspring* tetap layak untuk menjadi solusi. Masing-masing *offspring* dihitung *fitness*-nya kembali untuk penilaian kualitasnya. Contoh rancangan *crossover* dengan titik potong $a = 2$ dan $b = 6$ seperti yang telah diilustrasikan pada Gambar 5.



Gambar 5. Ilustrasi *crossover 2-point*

Metode mutasi menggunakan mutasi pada pengkodean nilai, yakni mengganti bagian gen yang posisinya didapatkan secara acak dengan indeks waktu ruang yang belum digunakan. Contohnya seperti gen : **2|20** yang belum memiliki ruang setelah kromosom *offspring* mengalami mutasi, kemudian dilakukan pengecekan kembali terhadap aturan umum. Jika terjadi pelanggaran terhadap aturan umum maka indeks ruang waktu diganti dengan indeks yang belum digunakan hingga didapatkan kromosom yang layak.

Pada proses *update* generasi dilakukan penggabungan kromosom *offspring* hasil proses mutasi dan kromosom hasil seleksi sebelumnya. Kemudian dari kumpulan kromosom tersebut dilakukan seleksi dengan menggunakan seleksi *roulette wheel* sebanyak jumlah kromosom pada populasi yang digunakan. Hasilnya berupa populasi yang berisi kromosom terpilih yang kemudian menggantikan populasi sebelumnya. *Update* generasi yang di ilustasikan pada Tabel 11. :

Tabel 11. Ilustrasi *update* generasi

Ruangan	Space	Senin								
		1	2	3	4	5	6	7	8	9
102	36		9 31					1 30		
103	55				7 30					

104	28			5 31						8 30
105	43		6 27				9 31			
106	40					4 22				
107	40								2 20	
201	40			3 25						
202	33						10 27			

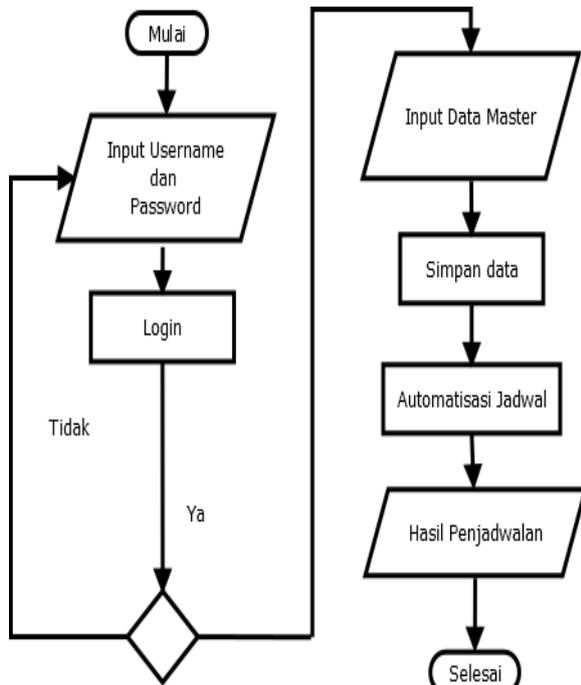
Setelah melakukan tahap 1 sampai 6 dalam Algoritma Genetika, maka akan menghasilkan generate jadwal perkuliahan. Dari hasil generate di atas maka didapatkan jadwal mata kuliah yang telah di pilih dengan generasi terbaik yaitu :

Tabel 12. Hasil jadwal generasi terakhir

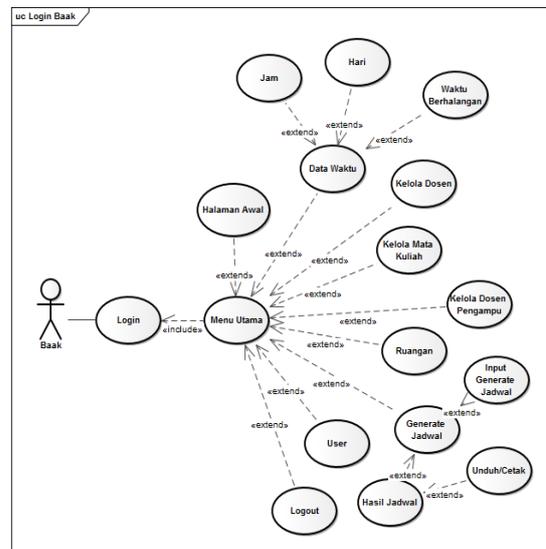
Hari	Sesi/Jam	Ruangan	Kd Matkul	Dosen	Kelas
Senin	08.15 - 09.35	102	M05	Kusnadi, M.Kom.	TI-SCN-1/6
	08.15 - 09.35	105	M03	Kusnadi, M.Kom.	TI-1/2
	09.15 - 10.35	104	M01	Ilwan Syafrinal, M.Kom.	TI-SCN-1/6
	09.15 - 10.36	201	M02	Ilwan Syafrinal, M.Kom.	TI-3/4
	10.15 - 11.35	103	M04	Kusnadi, M.Kom.	TI-SE-1/6
	11.15 - 12.35	106	M02	Ilwan Syafrinal, M.Kom.	TI-3/4
	12.15 - 13.35	105	M05	Kusnadi, M.Kom.	TI-SCN-1/6

2. Perancangan Sistem

Dalam perancangan *Activity diagram* menggambarkan bagaimana berjalannya sistem secara keseluruhan, yang di gambarkan sebagai berikut :



Gambar 6. Flowchart gambaran perancangan sistem



Gambar 7. Use case aktor BAAK

Keterangan Gambar 7 :

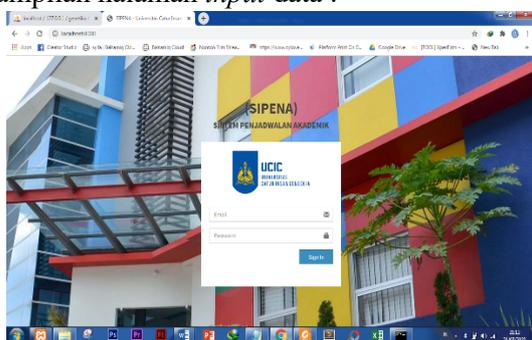
- Login : Sebelum masuk BAAK diharuskan memasukkan *username* dan *password*.
- Kelola data : BAAK mengelola master data seperti : data waktu, dosen, matakul, pengampu, ruangan dan penjadwalan.
- User : BAAK bisa melakukan perubahan password.
- Logout : BAAK bisa logout dari sistem agar tidak bisa di gunakan oleh orang yang tidak bertanggung jawab.

3. IMPLEMENTASI SISTEM

Tahap implementasi sistem adalah tahap sistem atau aplikasi yang telah di analisis dan dirancang. Dalam tahap implementasi sistem ini diperlukan spesifikasi kebutuhan sistem. Pada kebutuhan sistem ini akan dibahas mengenai perangkat keras, perangkat lunak,serta cara kerja dan penjelasan dari Sistem Penjadwalan Akademik yang akan dijelaskan dibawah ini:

a. Tampilan *Input*

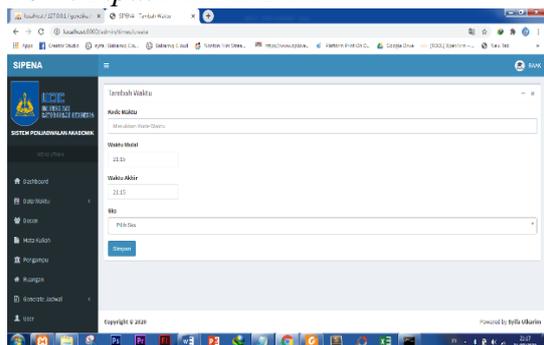
Tampilan Halaman *input* adalah halaman yang digunakan pengguna dalam memasukan data. Dibawah ini adalah tampilan halaman *input* data :



Gambar 8. Halaman login user baik

Pada halaman *form login* ini BAAK, Dekan, Dosen dan Mahasiswa memasukkan *username* dan *password* untuk bisa masuk ke halaman selanjutnya.

Form Input Master Data



Gambar 9. Halaman input master data

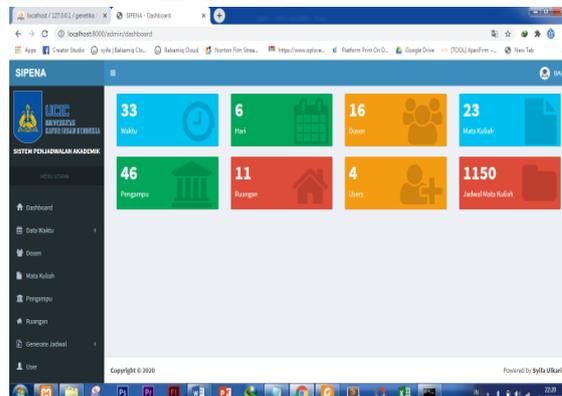
Halaman *form input* adalah halaman dimana BAAK menginputkan master data seperti : data jam, data hari, data waktu berhalangan, dosen, mata kuliah, dosen pengampu dan ruangan.

b. Tampilan *Output*

Tampilan output adalah halaman yang

digunakan pengguna untuk menampilkan data penjadwalan. Pada bagian ini dijelaskan halaman masing-masing level pengguna. Berikut adalah tampilan halaman masing-masing pengguna :

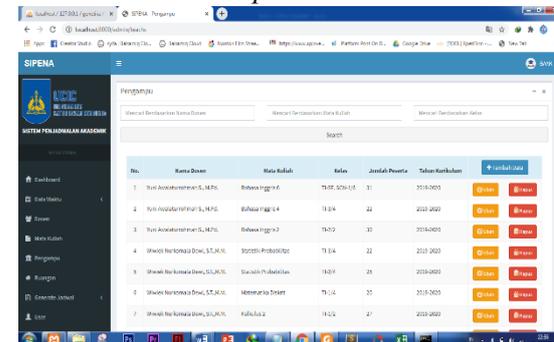
Halaman Utama BAAK



Gambar 10. Halaman utama BAAK

Pada halaman utama BAAK ini berisikan informasi mengenai jam, hari dan banyaknya ruangan, kelas, dosen, jadwal yang digunakan.

Contoh Halaman *Output*

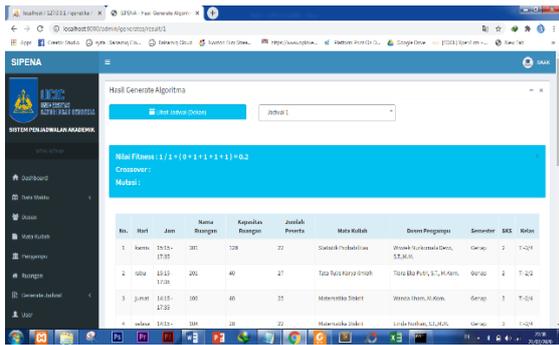


Gambar 11. Tampilan output Data Pengampu

Keterangan Gambar 11 :

Pada gambar yang di tunjukkan di atas adalah contoh *output* form isian dari data master. Yang di contohkan di atas adalah contoh *output* data pengampu.

Halaman *Output* Jadwal



Gambar 12. Tampilan output Data Jadwal
Keterangan Gambar 12 :

Pada gambar yang di tunjukkan di atas adalah contoh *output* dari hasil akhir *optimasi* penjadwalan kuliah.

4. HASIL PENGUJIAN DATA

Hasil pengujian data dilakukan untuk mengetahui persentase tingkat keberhasilan dari sistem yang di bangun. Data uji coba pada penjadwalan ini dengan menggunakan data jadwal kuliah TI semester genap T.A 2019/2020. Dengan rincian yang di tunjukkan pada Tabel 13. Di bawah ini :

Tabel 13. Hasil pengujian persentase *generate* jadwal

No.	Persentase Crossover	Persentase Mutasi	Jumlah Max Kromosom	Jumlah Max Generasi	Waktu Tunggu	Generasi						Total Error
						1	2	3	4	5	6	
1.	70%	70%	2	2	00:48:31	4	3	3	0	0	0	10
2.	80%	80%	3	3	02:20:46	3	3	0	0	0	0	6
3.	90%	90%	4	4	03:15:09	0	0	0	0	0	0	0
4.	100%	100%	5	5	04:35:15	1	6	3	1	0	0	11

Dari hasil uji coba, perbandingan waktu tunggu rata-rata adalah 00:04:36 menit . Pada 4 sampel uji coba, pada uji coba ke-1. Generasi pertama terdapat 4 bentrok (jumlah kapasitas ruangan lebih kecil daripada jumlah peserta mata kuliah) dan menghasilkan 4 generasi untuk mendapatkan jadwal yang optimal, sedangkan pada uji coba ke-2 didapatkan 3 generasi untuk mendapatkan jadwal yang optimal. Dan pada uji coba ke-3 didapatkan 1 generasi untuk mendapatkan jadwal yang optimal. Namun pada pengujian ke-4 tingkat akurasi *error*, pada generasi menjadi tidak akurat dan waktu proses *generate* jadwal yang paling lama. Ini dikarenakan masing-masing generasi saling merebutkan tingkat ke-akurasi yang tinggi yaitu 100% dan mencapai jadwal optimal pada generasi 5. Dalam uji coba yang telah di lakukan didapatkan untuk menghasilkan jadwal yang optimal dengan generasi yang sedikit yaitu dengan persentase 90%. Disini dapat di simpulkan :

1. Semakin banyak persentase maka akan semakin kecil generasi yang di dapatkan untuk menghasilkan jadwal yang optimal namun membutuhkan waktu lama.
2. Semakin banyak kelas yang di dijadwalkan maka akan semakin lama waktu yang di butuhkan oleh sistem genetika dalam menghasilkan jadwal kuliah.
3. Total percobaan 1-3 untuk menghasilkan

jadwal yang optimal dengan generasi yang sedikit membutuhkan presentase 90% dengan waktu 00:06:23 menit.

IV. KESIMPULAN DAN SARAN

Pada penelitian yang telah menghasilkan sistem penjadwalan kuliah dengan algoritma genetika ini dapat menghasilkan beberapa kesimpulan sebagai berikut:

1. Sistem mampu menghasilkan penjadwalan kuliah, dan mampu melakukan optimasi penjadwalan dengan hasil yang lebih efisien sehingga dapat memudahkan tugas BAAK Universitas Catur Insan Cendekia.
2. Sistem yang dibangun berhasil mengimplementasikan algoritma genetika untuk optimasi penjadwalan kelas mata kuliah di Universitas Catur Insan Cendekia.
3. Semakin besar persentase maka akan semakin kecil generasi yang di dapatkan untuk menghasilkan jadwal yang optimal namun membutuhkan waktu lama.
4. Semakin banyak kelas yang di dijadwalkan maka akan semakin lama waktu yang di butuhkan oleh sistem genetika dalam menghasilkan jadwal kuliah.

DAFTAR PUSTAKA

- [1] G. Studi, K. Di, A. Puspasari, K. Novianingsih,

- and F. Agustina, “Penyelesaian Masalah Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus Di Departemen Pendidikan Matematika Fpmipa Universitas Pendidikan Indonesia),” *Penyelesaian Masalah Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus Di Departemen Pendidikan Matematika Fpmipa Universitas Pendidikan Indonesia)*, vol. 7, no. 1, pp. 80–92, 2019.
- [2] S. Informatika *et al.*, “FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS HASYIM ASY ’ ARI Muhammad Ilham Zulhardani Bambang Sujatmiko Arbiati Faizah,” pp. 1–9, 2020.
- [3] A. Laksono, M. Utami, and Y. Sugiarti, “Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran Dan Kesehatan Universitas Muhammadiyah Jakarta),” *Studia Informatika: Jurnal Sistem Informasi*, vol. 9, no. 2, pp. 177–188, 2016, doi: 10.15408/sijisi.v9i2.7647.
- [4] E. B. Prasetyo, “Implementation of Genetic Algorithm and Artificial Neural Network In Lecture Scheduling In Faculty of Mathematics and Natural Science Universitas Gadjah Mada,” pp. 1–138, 2014.
- [5] E. Suhartono, “Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang),” *Infokam*, vol. 2, pp. 132–146, 2015.