

APLIKASI PENJADWALAN MATA KULIAH JURUSAN TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS PALANGKA RAYA DENGAN ALGORITMA GENETIKA

Ade Chandra Saputra^{a,1,*}, Agus Sehatman Saragih^{b,2}

^a Universitas Palangka Raya, Jl. H. Timang

^b Universitas Palangka Raya, Jl. H. Timang

¹ adechandra@it.upr.ac.id *; ² assaragih@gmail.com ;

* corresponding author

ARTICLE INFO

Keywords

Algoritma Genetika
Scheduling courses

ABSTRACT

Scheduling courses in informatics engineering is needed to arrange an optimal schedule so that there is no conflict when teaching courses. In this journal of informatics engineering has a problem in making a schedule because of frequent clash of teaching hours and the room used. Based on the problem, then to solve the above problem is to use the genetic algorithm method. The genetic algorithm method is used for scheduling optimization.

The final result of this research is the website program of Scheduling Courses in the Department of Informatics, Faculty of Engineering, Palangkaraya University, which can display information about course schedules and this will be a tool for departments to process scheduling data through the course scheduling application.

1. Pendahuluan

Salah satu masalah penjadwalan yang umum dijumpai di perguruan tinggi adalah masalah penjadwalan mata kuliah. Penjadwalan mata kuliah merupakan proses penyusunan jadwal pelaksanaan yang menginformasikan sejumlah mata kuliah yang diberikan, dosen yang mengajar, ruang tempat belajar, waktu serta mahasiswa yang mengambil mata kuliah tersebut. Di setiap semester, beberapa universitas menghadapi permasalahan yang sama, yaitu bagaimana menjadwalkan mata kuliah dengan kendala waktu, dosen yang tersedia, kapasitas ruangan yang terbatas, dan mahasiswa yang mengambil mata kuliah, dimana proses penjadwalan tersebut tetap memperhatikan sejumlah batasan dan syarat-syarat tertentu.

Untuk menyusun jadwal mata kuliah yang baik kita perlu memperhatikan berbagai aspek yang mempengaruhi penjadwalan mata kuliah. Dari aspek dosen, kita perlu memperhatikan kemungkinan dosen yang bersangkutan tidak dapat mengajar karena kegiatan akademik yang lain, dosen mengajar lebih dari satu mata kuliah pada hari dan jam yang sama serta dosen sudah memiliki jadwal mengajar di jurusan lain. Peneliti juga perlu memperhatikan dari aspek mahasiswa ada atau tidaknya jadwal mata kuliah yang beririsan dengan jadwal kuliah angkatan sebelumnya maupun sesudahnya sehingga mahasiswa dapat mengambil mata kuliah angkatan sebelumnya maupun sesudahnya. Selain itu, perlu diperhatikan juga dari aspek ketersediaan ruangan belajar dan ruang praktikum pada mata kuliah yang berpraktikum agar proses perkuliahan dapat terlaksana. Berdasarkan aspek-aspek di atas, dalam penyusunan jadwal kuliah terdapat banyak kemungkinan yang dapat dicoba untuk menemukan penjadwalan yang baik.

Oleh karena itu dibutuhkan metode optimasi yang dapat diterapkan untuk mengerjakan penjadwalan mata kuliah.

Masalah optimasi dapat diselesaikan dengan beberapa cara, salah satunya dengan menggunakan algoritma pencarian heuristik. Untuk kasus yang sederhana digunakan algoritma pencarian heuristik yang biasa seperti *breadth-first search and depth-first search*. Untuk input dan persyaratan yang lebih rumit seperti pada kasus penjadwalan mata kuliah, algoritma pencarian heuristik sudah tidak dapat digunakan dengan baik untuk mendapatkan solusi yang diinginkan. Dalam kasus penjadwalan mata kuliah, diperlukan algoritma yang lebih baik yaitu algoritma yang dapat menyelesaikan masalah multi-kriteria dan multi-objektif. Salah satu algoritma yang dapat digunakan adalah algoritma genetika.

Ada tiga keunggulan dari aplikasi algoritma genetika dalam proses optimasi (Sanjoyo, 2006) yaitu:

- a. Algoritma genetika tidak terlalu banyak memerlukan persyaratan matematika dalam penyelesaian proses optimasi. Algoritma genetika dapat diaplikasikan pada beberapa jenis fungsi obyektif dengan beberapa fungsi pembatas baik berbentuk linier maupun non-linier.
- b. Operasi evolusi dari algoritma genetika sangat efektif untuk mengobservasi posisi global secara acak.
- c. Algoritma genetika mempunyai fleksibilitas untuk diimplementasikan secara efisien pada problematika tertentu.

Proses evolusi menjadi ciri spesifik dari algoritma genetika, suatu proses seleksi alamiah yang berdasarkan prinsip genetika dan pemilihan alami pada individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya, sehingga terseleksi individu-individu yang kuat yang mampu bertahan sebagai hasil optimal (Basuki, 2003; Haupt, 2004). Algoritma genetika dapat digunakan untuk menyelesaikan permasalahan optimisasi yang rumit (*hard optimization problems*) yang tidak dapat diselesaikan dengan teknik optimasi tradisional seperti masalah penjadwalan, transportasi, permainan komputer, *traveling salesman problem* (TSP), dan lain-lain.

Perbedaan algoritma genetika dengan algoritma pencarian lainnya, antara lain (Goldberg, 1989):

- a. Algoritma genetika secara langsung bekerja dengan string yang mempresentasikan kumpulan parameter.
- b. Algoritma genetika menggunakan keputusan probabilistik (non deterministik) dalam pencarian solusinya.

Ciri-ciri permasalahan yang dapat dikerjakan dengan menggunakan algoritma genetika adalah (Basuki, 2003):

- a. Mempunyai fungsi tujuan optimalisasi non linear dengan banyak kendala yang juga non linear.
- b. Mempunyai kemungkinan solusi yang jumlahnya banyak.
- c. Membutuhkan solusi "*real-time*" dalam arti solusi dapat diperoleh dengan cepat.
- d. Mempunyai multi-objektif dan multi-kriteria.

Algoritma genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Diharapkan dengan digunakannya algoritma genetika akan diperoleh optimasi penjadwalan mata kuliah dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, serta tidak ada permasalahan jadwal yang beririsan.

Dari permasalahan diatas maka dibuatlah "Aplikasi Penjadwalan Matakuliah di Jurusan Teknik Informatika menggunakan Algoritma Genetika". Diharapkan dengan sistem tersebut

dapat membantu pembuatan jadwal mata kuliah pada Jurusan Informatika Fakultas Teknik Universitas Palangkarya.

1.1 Kajian Literatur Dan Pengembangan Hipotesis

Optimasi adalah salah satu disiplin ilmu dalam matematika dengan tujuan mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang, maupun pencarian nilai dalam berbagai kasus. Aplikasi dari teknik optimasi telah banyak pada berbagai bidang secara cepat. Keberhasilan penerapan teknik optimasi memerlukan tiga syarat, yaitu: kemampuan membuat model matematika dari permasalahan yang dihadapi, pengetahuan akan program komputer dan pengetahuan akan teknik optimasi itu sendiri.

Masalah optimasi dapat dikategorikan ke dalam dua kelas besar, yaitu optimasi tanpa pembatas (*unconstrained optimization*) dan optimasi dengan pembatas (*constrained optimization*). Optimasi tanpa pembatas merupakan optimasi yang hanya melibatkan fungsi tujuan, tidak ada pembatas (*constraint*), sedangkan optimasi dengan pembatas, selain fungsi tujuan juga mempunyai tambahan pembatas yang membuat permasalahan lebih rumit. Dalam *constrained optimization*, dengan adanya pembatas diperlukan algoritma yang berbeda untuk menyelesaikannya.

Prosedur pemecahan masalah optimasi dilakukan dengan cara memodelkan persoalannya ke dalam sebuah program matematis dan kemudian memecahkannya dengan menggunakan teknik-teknik atau metode optimasi seperti program linier, program nonlinier, program tujuan ganda, dan metode-metode lainnya yang sudah berkembang saat ini. Dalam penelitian ini akan menggunakan metode algoritma genetika untuk menyelesaikan persoalan optimasi. Permasalahan optimasi telah banyak diselesaikan dengan menggunakan algoritma genetika, dan hasil yang diperoleh biasanya lebih baik, walaupun tidak menutup kemungkinan bahwa ada juga hasil akhir yang kurang begitu memuaskan (Desiani, 2006).

Penjadwalan merupakan proses untuk menyusun suatu jadwal atau urutan proses yang diperlukan dalam sebuah permasalahan. Permasalahan penjadwalan biasanya berhubungan dengan penjadwalan kelas dalam sekolah atau perkuliahan dan juga dalam lingkup yang tidak jauh berbeda seperti penjadwalan pelajaran sekolah, penjadwalan ujian, atau dapat juga penjadwalan karyawan, baik dalam suatu perusahaan ataupun dalam rumah sakit. Dalam penjadwalan kuliah, yang dibahas adalah tentang pembagian jadwal untuk tiap mahasiswa pada kuliah tertentu sekaligus dosen pengajarnya. Dalam penjadwalan pelajaran sekolah yang dibahas adalah tentang pembagian jadwal pelajaran untuk tiap-tiap kelas yang ada beserta guru pengajar pelajaran tersebut, sedangkan dalam penjadwalan ujian yang dibahas adalah tentang pengaturan dosen yang mengawasi ujian dan mahasiswa yang menempati ruang.

Terdapat dua batasan dalam penyusunan penjadwalan kuliah yang dikemukakan oleh Burk dkk (1997), yaitu :*hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi).

Hard constraints merupakan batas-batas yang harus diterapkan pada penjadwalan mata kuliah dan harus dipenuhi. Sebuah solusi hanya dapat dikatakan sah dan valid apabila dalam solusi tersebut sama sekali tidak ada *hard constraint* yang terlanggar. *Hard constraints* yang umum dalam penjadwalan mata kuliah adalah sebagai berikut :

- a. Seorang dosen hanya dapat mengajarkan mata kuliah untuk satu lokasi pada waktu tertentu.
- b. Seorang mahasiswa hanya dapat mengikuti kuliah untuk satu lokasi pada waktu tertentu.
- c. Sebuah lokasi (ruangan) hanya dapat digunakan untuk satu mata kuliah pada waktu tertentu.

- d. Mata kuliah dengan bobot 3 SKS dijadwalkan dengan satu kali pertemuan dalam seminggu.
- e. Hari aktif untuk perkuliahan adalah hari Senin sampai dengan Sabtu.

Berbeda dengan *hard constraint*, *soft constraint* merupakan kendala yang tidak selalu dapat terpenuhi dalam proses pembentukan jadwal, akan tetapi meskipun tidak harus terpenuhi, jadwal yang dihasilkan harus semaksimal mungkin berusaha memenuhi ketentuan *soft constraint*. Contoh *soft constraints* dalam penjadwalan mata kuliah antara lain :

- a. Dosen dapat meminta waktu mengajar tertentu yang diinginkan.
- b. Penempatan jadwal untuk waktu yang telah diminta dosen disesuaikan dengan prioritas dosen.

Algoritma genetika pertama kali ditemukan oleh John Holland pada tahun 1975 melalui sebuah penelitian dari Universitas Michigan yang kemudian dikembangkan oleh muridnya David Goldberg pada tahun 1989. Awalnya tujuan Holland adalah bukan untuk merancang algoritma yang mampu memecahkan masalah yang spesifik, melainkan hanya untuk mempelajari fenomena adaptasi seperti yang terjadi di alam sekaligus untuk mengembangkan cara-cara bagaimana mekanisme adaptasi alam dapat diimpor ke sebuah sistem komputer. John Holland dalam bukunya yang berjudul *Adaptation In Natural And Artificial Systems*, menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Untuk memahami algoritma genetika diperlukan 3 tahapan pendalaman yaitu: aspek filosofis, struktur dan komponen.

Menurut Lawrence (1991), Algoritma Genetika adalah algoritma kecerdasan buatan tentang teknik pencarian dan optimasi yang berdasarkan pada mekanisme seleksi atau evolusi yang terjadi di alam. Algoritma genetika merupakan salah satu cabang dari algoritma evolusi yang digunakan untuk memecahkan masalah optimasi yang kompleks dan sulit diselesaikan dengan menggunakan metode yang konvensional (tradisional). Algoritma ini didasarkan pada proses genetika yang ada dalam makhluk hidup; yaitu perkembangan populasi yang alami, secara lambat laun mengikuti proses seleksi alam atau Siapa yang kuat, dia yang bertahan (*survive*). Prinsip pengerjaan algoritma genetika adalah *Survival Of The Fittest* (kemampuan kelangsungan hidup) dengan pemecahan yang berbeda dalam sebuah solusi. Pemecahan terbaik yang diperoleh dari bentuk baru diharapkan merupakan pemecahan yang terbaik dari populasi, sedangkan pemecahan yang kurang baik dihilangkan. *Survival* dalam algoritma genetika merupakan suatu pewarisan genetik yang dijadikan sebagai model metode pencarian pada suatu fenomena alam. Dengan meniru teori evolusi, algoritma genetika dapat digunakan untuk mencari solusi yang tepat pada permasalahan-permasalahan optimasi dalam dunia nyata baik satu variabel maupun multivariabel seperti pada permasalahan penjadwalan, *Travelling Salesman Problem* (TSP), perancangan *neural networks* (jaringan syaraf tiruan), dan *Job Shop Scheduling*.

2. Metode Penelitian

1. Pengumpulan Data

Tahap pengumpulan data ini mencari bahan dasar yaitu mengumpulkan data mata kuliah, data dosen dan data kelas di jurusan teknik informatika di bagian kemahasiswaan

Pengumpulan data merupakan awal dari penelitian, dalam hal ini pengumpulan data dapat meliputi :

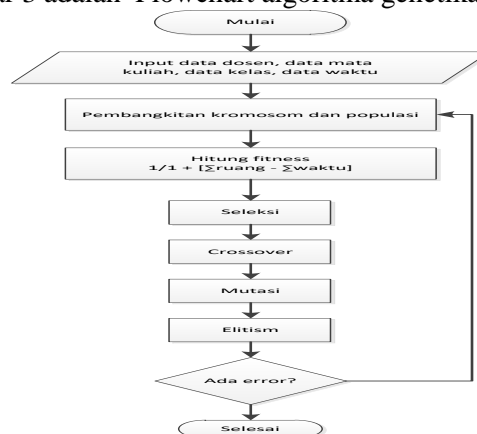
- a) Observasi

Observasi yaitu melakukan pengamatan secara langsung ke tempat objek yang diamati secara dekat tentang masalah yang dihadapi dan melakukan pencatatan temuan secara khusus

- b) Wawancara



- Melakukan wawancara langsung dengan pihak terkait di jurusan teknik informatika
2. Studi Pustaka
Studi pustaka dalam penelitian ini dapat dilakukan dengan mempelajari beberapa literature, misalnya melalui media internet, jurnal, maupun buku – buku yang berhubungan dengan penelitian yang meliputi : penerapan konsep algoritma genetika.
 3. Pengolahan Data
Proses pengolahan data pada penelitian ini menggunakan algoritma genetika dimana memiliki beberapa tahapan. Tahapan dimulai dari proses seleksi, crossover, mutasi dan elitisme. Pada Gambar 3 adalah Flowchart algoritma genetika :

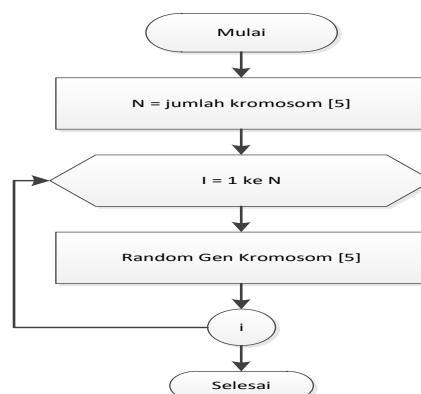


Gambar 1 Flowchart Algoritma Genetika yang akan diterapkan

Flowchart menggambarkan proses umum algoritma genetika dimana didalamnya terdapat beberapa sub proses meliputi

a. Pembangkitan kromosom dan populasi

Proses pembangkitan populasi dilakukan secara acak sejumlah dengan individu atau solusi yang diinginkan. Diasumsikan dalam satu jadwal ada 4 gen dalam satu kromosom terdiri dari mata kuliah, kelas, ruang dan waktu. Penjadwalan dilakukan untuk kurun waktu 5 hari sehingga memiliki 5 kromosom. Proses dilakukan secara acak dengan proses seperti gambar flowchart gambar 4.



Gambar 2 Flowchart Pembangkitan Kromosom dan Populasi

Random akan terbentuk terdiri dari sekumpulan gen yang mewakili mata kuliah, kelas, ruang dan waktu. 1 individu berlaku untuk 5 hari sehingga akan terbentuk acak kromosom seperti berikut :

M02K01W01 M05K02W03 M01K01W03 M06K02W04 M09K01W05

b. Fitness

Nilai fitness diambil dari rash antar jadwal selama algoritma dijalankan. Rumus fitness ditunjukkan pada rumus 1

$$fitness = \frac{1}{1 + \sum rash} \dots\dots\dots(1)$$

Poin – poin crash yang tidak boleh terjadi yaitu

- Crash dosen dan mata kuliah yang berlangsung
- Crash kelas yang digunakan
- Crash waktu yang berlangsung

Berdasar constraint yang sudah di dapat sehingga rumus fitness untuk penelitian ini yaitu

$$fitness = \frac{1}{1 + (\sum c.kelas + \sum c.waktu)}$$

Contoh penerapan fitness pada kromosom. Missal terjadi crash pada waktu.

$$Fitness = \frac{1}{1 + (0 + 1)} = \frac{1}{2} = 0.5$$

c. Seleksi

Proses seleksi dilakukan untuk mendapatkan individu terbaik sehingga di dapat solusi yang baik untuk hasil akhir. Proses seleksi menggunakan metode roulette wheel. Proses roulette wheel dijabarkan seperti berikut. Asumsikan fitness dari kelima kromosom memiliki nilai sesuai table 2 berikut

Tabel 1 fitness

Kromosom	Fitness
1	0.5
2	0.3
3	0.5
4	0.3
5	0.25
Total Nilai Fitness	1.85

Tahap selanjutnya yaitu mencari probabilitas sesuai dengan table 3

Tabel 2 probabilitas

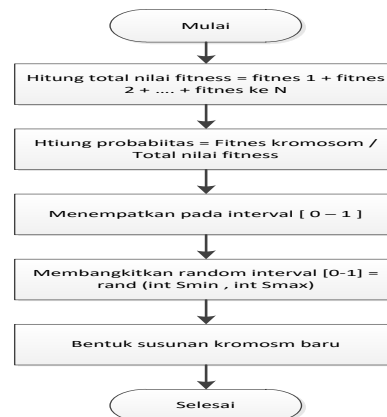
Kromosom	Probabilitas
1	0.5 / 1.85 = 0.27
2	0.3 / 1.85 = 0.16
3	0.5 / 1.85 = 0.27
4	0.3 / 1.85 = 0.16
5	0.25 / 1.85 = 0.13
Total Probabilitas	1

Proses selanjutnya membuat interval setiap probabilitas untuk proses pembangkitan. Interval pada table 4

Tabel 3 interval setiap probabilitas

Kromosom	Interval
1	0 - 0.27
2	0.27 - 0.43
3	0.43 - 0.7
4	0.7 - 0.86
5	0.86 - 1

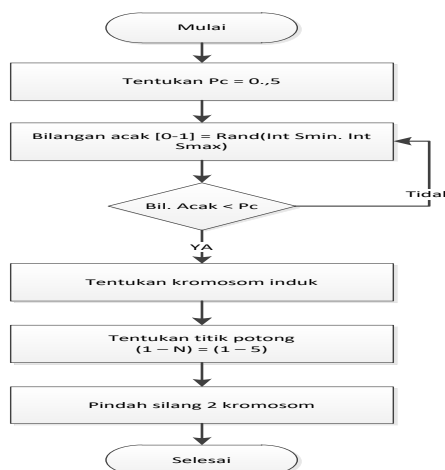
Pembangkitan kromosom kemudian random interval di dalam table. Apabila menghasilkan nilai random (0.71 : 0.3 : 0.9 : 0.1 : 0.54) maka sususna kromosom baru yaitu kromosm 4, kromosom 2, kromosom 5, kromosom 1 dan kromosom 3. Tahapan proses seleksi dijelaskan pada gambar 5 flowchart berikut



Gambar 3 Tahapan proses seleksi

d. Crossover

Proses crossover merupakan proses penggabungan antar kromosom. Adapun flowchart crossover sesuai gambar 6 berikut



Gambar 4 Proses crossover

Nilai probabily crossover (Pc) sudah ditentukan sebesar 0.5. Random bilangan dengan interval [0-1] didapat dari nilai probabilitas setiap kromosom. Perhitungan probabilitas

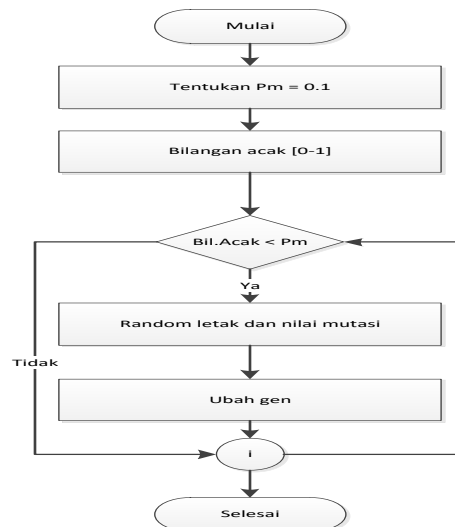


di dapat dari proses seleksi sebelumnya. Langkah berikutnya yaitu menentukan kromosom induk dengan mengambil hasil perulangan bilangan acak yang memiliki nilai kurang dari P_c yaitu sebesar 0.5.

Tahap selanjutnya yaitu mencari titik potong. Rumus mencari titik potong yaitu $[1 - N]$ dimana N adalah jumlah gen dalam satu kromosom. Apabila diasumsikan nilai $N= 5$, maka titik potong berada pada $[1-5] = 4$ atau gen ke 4. Langkah terakhir yaitu melakukan crossover dua kromosom sesuai syarat sebelumnya yaitu kromosom dengan probabilitas dibawah $P_c = 0.5$ dan letak titik potong ada pada gen ke 4.

e. Mutasi

Proses mutasi merupakan proses merubah gen dalam satu kromosom. Batasan yang ada pada mutasi yaitu probability mutation (P_m) dan memiliki interval $[0-1]$. P_m pada kasus ini di set sehingga memiliki nilai 0.1. hal ini dilakukan supaya selama proses mutasi tidak terjadi perubahan major pada nilai kromosom atau bisa dicek pada nilai fitness kromosom. Berikut gamabr 7 proses mutasi



Gambar 5 Proses mutasi

Proses selanjutnya yaitu melakukan pencarian kromosom yang memiliki nilai dibawah P_m . Apabila menemukan kromosom yang memiliki nilai dibawah P_m maka melakukan random letak dan nilai untuk proses mutasi. Iterasi proses mutasi hingga tidak ada nilai probabilitas di bawah nilai P_m

f. Elitism

Tahapan terakhir dari sub proses algoritma genetika adalah elitism. Elitism merupakan proses menyimpan individu terbaik sehingga menghasilkan solusi yang optimal. Dalam proses elitism dibagi menjadi dua kemungkinan yaitu apabila terjadi crash pada jadwal maka nilai fitness buka sama dengan 1 dan apabila tidak ada crash maka nilai fitness sama dengan 1.

4. Analisa Kebutuhan Output

Setelah proses inputan diproses akan menghasilkan berupa output jadwal perkuliahan .

5. Perancangan

Sistem yang akan dibangun tersebut terlebih dahulu perlu dilakukan suatu perancangan yang mana perancangan tersebut meliputi :

a) Perancangan Proses



Rancangan proses dibutuhkan untuk mengetahui alur pemrosesan yang dilakukan dalam bentuk sebuah diagram alir atau *flowchart*. Alur sistem yang akan dibuat ini hanya terdapat 2 entitas yaitu admin (kepala jurusan) dan staff jurusan yang dibuat dalam bentuk *flowchart*, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD).

b) Perancangan Database

Perancangan database atau basis data dibuat untuk penyimpanan data yang dibutuhkan dalam sistem yang akan dibangun.

c) Perancangan Struktur Menu

Perancangan struktur ini dibuat untuk mengetahui daftar menu yang terdiri atas proses-proses yang ada didalam sistem yang akan dibangun.

d) Perancangan Antar Muka (*user interface*)

Rancangan *user interface* atau antar muka bermanfaat dalam mengawali perancangan tampilan. Dengan adanya perancangan *user interface* dan prosedural, pengimplementasian sistem lebih mudah dilakukan. Selain itu di dalam perancangan antar muka ini akan menggambarkan stuktur menu dari tahapan proses yang terdapat pada sistem.

6. Analisa Implementasi dan Pengujian

Pada tahap ini dimulai dengan proses pemilihan perangkat keras (*hardware*), *coding* atau implementasi sistem yang akan dibangun, serta sistem pengujian.

7. Hasil dan Kesimpulan

Setelah melakukan evaluasi maka dapat ditentukan hasilnya berdasarkan hasil evaluasi dari pengujiannya. Kesimpulan dapat bernilai positif dan negatif, hal ini sesuai dengan hasil yang diperoleh pada evaluasi.

3. Hasil Dan Pembahasan

Implementasi sistem adalah penerapan hasil perancangan yang telah diuraikan sebelumnya pada bab perancangan sistem. Implementasi yang dilakukan antara lain adalah menerapkan perancangan antarmuka ke dalam bentuk software. Tujuan implementasi adalah mengkonfirmasi model program perancangan pada sistem sehingga user dapat memberikan masukan kepada pembangun sistem. Berikut ini merupakan gambaran tampilan antarmuka yang telah dibuat

3.1 Implementasi Proses

1. Tampilan halaman website

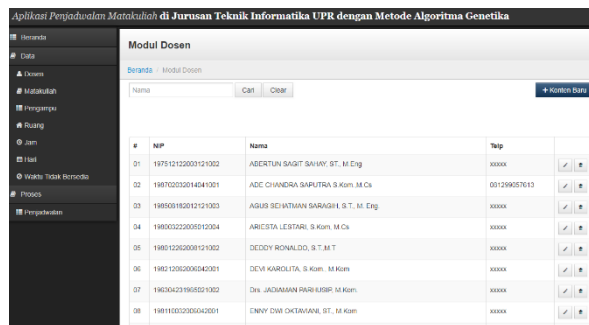
Pada proses ini, user yang akan menggunakan aplikasi dengan menginputkan data dosen, data matakuliah, data pengampun matakuliah, data ruang, data jam, data hari dan waktu yang tidak tersedia dalam artian rule dosen tidak bisa mengajar pada jam dan hari apa saja. Setelah selesai penginputan data semua langkah berikutnya yaitu melakukan penyusunan jadwal menggunakan algoritma genetika.



Gambar 8 Tampilan Halaman Web

2. Tampilan halaman input data dosen

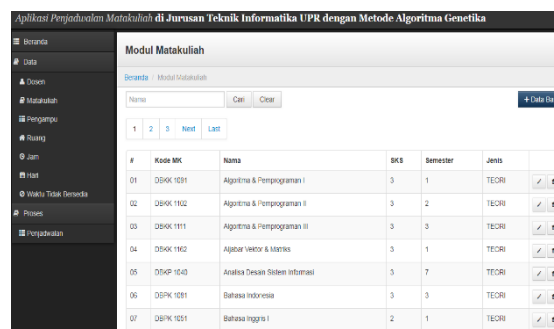
Pada tampilan ini halaman input dosen ini user memasukan data – data dosen



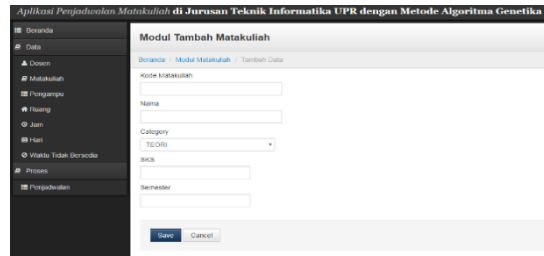
Gambar 9 Tampilan Halaman Dosen

3. Tampilan Halaman Input Matakuliah

Halaman ini digunakan untuk menginputkan matakuliah apa saja yang ada di teknik informatika beserta semester Berikut tampilannya

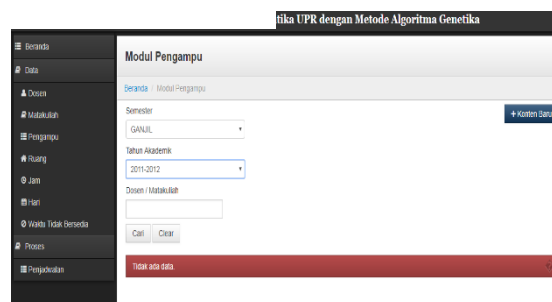


Gambar 10 Tampilan Halaman Matakuliah

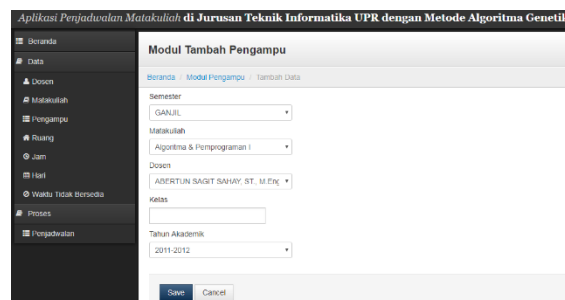


Gambar 11 Tampilan Halaman Input Matakuliah

4. Tampilan Halaman Pengampu



Gambar 12 Halaman Pengampu Matakuliah



Gambar 13 Tampilan Halaman Input Pengampu Matkul

5. Tampilan Halaman Ruang

Aplikasi Penjadwalan Matakuliah di Jurusan Teknik Informatika UPB dengan Metode Algoritma Genetika

Modul Ruang

#	Nama	Kapasitas	Jenis	
01	Ruang T-1	40	TEORI	
02	Ruang T-2	40	TEORI	
03	Ruang T-3	40	TEORI	
04	Ruang T-4	40	TEORI	
05	Ruang T-5	40	TEORI	

Gambar 13 Tampilan Halaman ruang kelas

6. Tampilan Halaman Jam

Aplikasi Penjadwalan Matakuliah di Jurusan Teknik Informatika UPB dengan Metode Algoritma Genetika

Modul Jam

#	Rangae Jam	
01	07.00 - 07.50	
02	07.50 - 08.40	
03	08.40 - 09.30	
04	09.30 - 10.20	
05	10.20 - 11.10	
06	11.10 - 12.00	
07	12.00 - 12.50	

Gambar 14 Tampilan Halaman Jam Ajar

7. Tampilan Halaman Hari

Aplikasi Penjadwalan Matakuliah di Jurusan Teknik Informatika UPB dengan Metode Algoritma Genetika

Modul Hari

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

Gambar 15 Tampilan Halaman Hari Kuliah

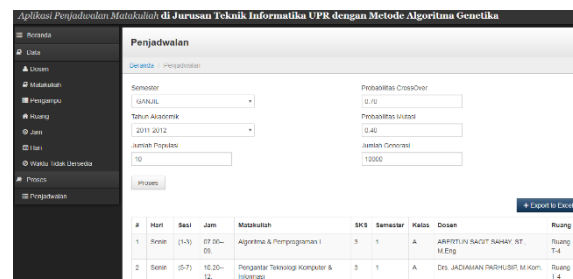
8. Tampilah Halaman Waktu Tidak Tersedia



Gambar 16 Tampilan Halaman Waktu Tersedia

9. Tamplina Halaman Penjadwalan

Pada Tampilan halaman penjadwalan ini di kerjakan dengan algoritma genetika , user diminta menginputkan jumlah populasi, probabilitas crossover, probabilitas mutasi, jumlah generasi, semester berapa akan digunakan, serta tahun akademik.



Gambar 17 Tampilan Halaman Penjadwalan

a. Pengujian Sistem

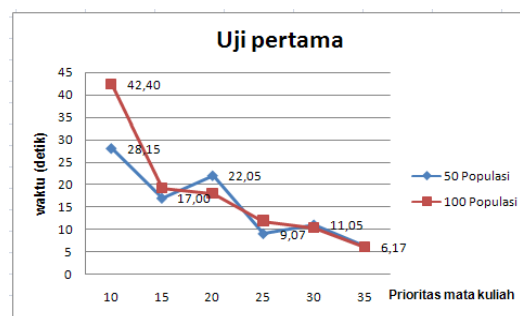
Pengujian parameter algoritma genetika ini bertujuan untuk menganalisa kinerja algoritma genetika pada proses penyusunan jadwal mata kuliah sesuai dengan ketentuan yang telah ditetapkan pengujian nantinya dilakukan dengan cara mengubah-ubah masukan parameter pada proses penjadwalan sehingga nanti diharapkan bisa didapat rekomendasi atau patokan parameter algoritma genetika yang menghasilkan proses penyusunan jadwal matakuliah yang baik.

Pada table 15 dibawah ini menunjukkan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pc dengan nilai 75 dan probabilitas mutasi Pm dengan nilai 3. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 4 Hasil pengujian pertama

Populasi	Pc	Pm	Fitness	Aktivitas Perkuliahan			Waktu (detik)
				Jum Mtkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	3	90,909	10	1	3	28,15
50	75	3	93,75	15	2	1	17,00
50	75	3	90	20	1	2	22,05
50	75	3	96	25	1	2	9,07
50	75	3	96,774	30	0	3	11,05
50	75	3	97,222	35	0	0	6,17
100	75	3	90,909	10	0	3	42,40
100	75	3	93,75	15	0	2	19,21
100	75	3	95,238	20	1	2	17,99
100	75	3	96,153	25	0	2	11,89
100	75	3	96,774	30	2	3	10,33
100	75	3	97,222	35	1	2	6,06

Dari data table 15 terlihat bahwa jadwal terbaik diperoleh pada masukan 35 matakuliah dan jumlah populasi 50 dimana hanya satu frekuensi dosen yang tinggi dalam sehari dan tidak terdapat frekuensi kelas yang tinggi dalam sehari dengan waktu pengekseskuan 6,17 detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 25 matakuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat dua frekuensi kelas yang tinggi dan waktu pengekseskuan 11,89 detik.



Gambar 18 Grafik pengujian Pertama

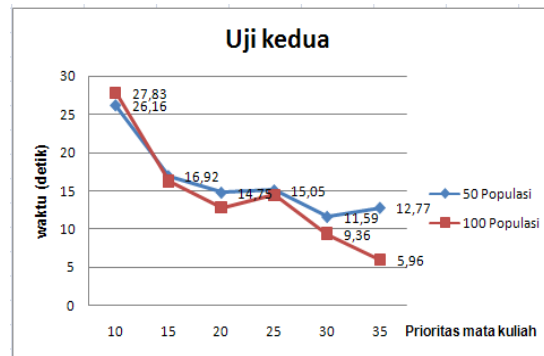
Pada gambar 18 menunjukkan laju kecepatan algoritma genetik dalam mengekseskusi, bisa terlihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengekseskuan yang lebih lama. Pada table 16 dibawah ini menunjukkan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pcdengan nilai75 dan probabilitas mutasi Pm dengan nilai4. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 5 Hasil pengujian kedua

Populasi	Pc	Pm	Fitness	Aktivitas Perkuliahan			Waktu (detik)
				Jm Mtkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	4	83,333	10	0	3	26,16
50	75	4	79,947	15	2	2	16,92
50	75	4	90,909	20	0	2	14,75
50	75	4	96,153	25	0	3	15,05
50	75	4	96,774	30	1	2	11,59
50	75	4	97,142	35	1	2	12,84
100	75	4	90,909	10	2	3	27,83
100	75	4	88,235	15	0	0	16,25
100	75	4	90	20	0	2	12,84
100	75	4	96,153	25	0	1	14,47
100	75	4	96,774	30	1	1	9,36
100	75	4	97,222	35	0	2	5,96

Dari data table 16 terlihat bahwa jadwal terbaik diperoleh pada masukan 20 mata kuliah dan jumlah populasi 50 dimana tidak terdapat frekuensi dosen yang tinggi dalam sehari dan terdapat dua frekuensi kelas yang tinggi dalam sehari dengan waktu pengekseskuan 14,75

detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 25 matakuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat satu frekuensi kelas yang tinggi dan waktu pengekseskuan 14,75 detik.



Gambar 19 Grafik Uji kedua

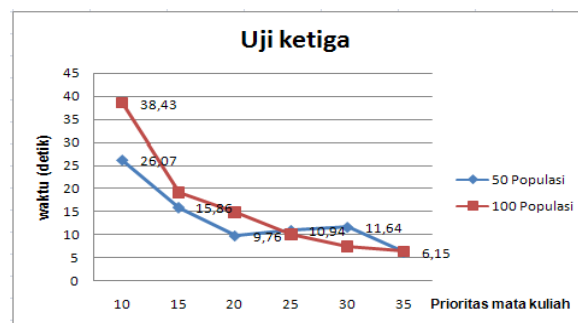
Pada gambar 19 menunjukkan laju kecepatan algoritma genetic dalam mengekseskusi data, bisa terlihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengekseskuan yang lebih lama.

Pada table 17 dibawah ini menunjukkan beberapa hasil pengujian sistem algoritma genetika dengan mengubah-ubah parameter masukan dan probabilitas *crossover* Pc dengan nilai 75 dan probabilitas mutasi Pm dengan nilai 5. Apabila jumlah populasi diset 50 dan 100, jumlah generasi diset 100.

Tabel 6 Hasil pengujian ketiga

Populasi	Pc	Pm	Fitness	Aktivitas Perkuliahan			Waktu (detik)
				Jml Mitkul	Frekuensi Dosen	Frekuensi Kelas	
50	75	5	90,909	10	2	3	26.07
50	75	5	93,75	15	1	4	15.85
50	75	5	95	20	0	2	9.76
50	75	5	96,153	25	0	3	10.94
50	75	5	96,774	30	1	2	11.64
50	75	5	97,222	35	0	1	6.15
100	75	5	90,909	10	1	3	38.43
100	75	5	93,75	15	0	1	19.21
100	75	5	95	20	0	2	14.87
100	75	5	96,153	25	1	1	10.04
100	75	5	96,774	30	0	2	7.41
100	75	5	97,222	35	0	2	6.39

Dari data table 17 terlihat bahwa jadwal terbaik diperoleh pada masukan 35 mata kuliah dan jumlah populasi 50 dimana tidak terdapat frekuensi dosen yang tinggi dalam sehari dan terdapat satu frekuensi kelas yang tinggi dalam sehari dengan waktu pengekseskuan 6,15 detik, sedangkan pada populasi 100 jadwal yang terbaik pada masukan 15 matakuliah dimana tidak terdapat frekuensi dosen yang tinggi tetapi terdapat satu frekuensi kelas yang tinggi dan waktu pengekseskuan 19,21 detik.



Gambar 20 Grafik Uji ketiga

Pada gambar 20 menunjukkan laju kecepatan algoritma genetik dalam mengeksekusi data, bisa terlihat bahwa semakin besar ukuran populasi kemungkinan lebih besar untuk menemukan solusi yang terbaik, namun membutuhkan waktu pengekseskuan yang lebih lama. Pada cobaan di atas bisa dilihat bahwa dengan merubah parameter masukan dan parameter algoritma genetika seperti ukuran populasi, probabilitas cross over dan probabilitas mutasi menghasilkan data yang bervariasi baik dalam waktu pengekseskuan maupun nilai fitnessnya. Jadi ukuran parameter lebih besar belum tentu menghasilkan jadwal terbaik, hasil penjadwalan bisa dilihat pada tabel lampiran. Ukuran populasi dan generasi mempengaruhi untuk kerja yang baik dan keefektifan algoritma genetik. Algoritma genetik dengan populasi dan generasi yang kecil biasanya unjuk kerjanya buruk karena populasi tidak menyediakan cukup materi untuk mencakup ruang persoalan. Populasi yang lebih besar dibutuhkan untuk mempresentasikan keseluruhan ruang persoalan. Namun, semakin besar ukuran populasi dan generasi maka kecepatan eksekusi program algoritma genetik akan semakin lambat.

4 Kesimpulan

Kesimpulan yang dihasilkan dari penelitian ini :

1. Sistem penjadwalan mata kuliah telah berhasil membuat jadwal perkuliahan dengan menerapkan algoritma genetik dan menggunakan sampel data pada Jurusan Teknik Informatika.
2. Sistem penjadwalan mata kuliah dapat mengakomodir untuk jadwal mata kuliah reguler.
3. Sistem penjadwalan mata kuliah yang dibuat telah dapat menjamin pemilihan slot waktu mengajar bagi dosen

Daftar Pustaka

- [1] Al Huda, dkk. 2005. Peramalan time series saham menggunakan backpropagation neural network berbasis algoritma genetik, *Jurnal Teknik Informatika PTIIK Universitas Brawijaya*, Malang.
- [2] Fadliyah, dkk. 2009. *Algoritma Genetik*. Graha Ilmu. Yogyakarta.
- [3] Fausett, L. 1994. *Fundamental of Neural Network : Architecture, Algorithm and Application*. New Jersey. Prentice-Hall.
- [4] Firdaus, dkk. 2007. Penerapan Algoritma Genetik pada Pelatihan Jaringan Syaraf Tiruan untuk Peramalan Debit Inflow Waduk Senggruh. *Jurnal Teknik Informatika PTIIK Universitas Brawijaya*, Malang.
- [4] Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.

- [5] Neves, J, and Cortez, P. 1998. *Combining Genetic Algorithms, Neural Networks and Data Filtering for Time Series Forecasting*. Departamento de Informatica Universidade do Minho. Portugal.
- [6] Pandjaitan, L.W. 2002. *Dasar-Dasar Komputasi Cerdas*. Andi Offset. Yogyakarta.
- [7] Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Andi Offset. Yogyakarta.
- [8]Suyanto. 2005. *Algoritma Genetika dalam MATLAB*. Andi Offset. Yogyakarta.
- [9]Warsito, B. 2009. *Kapita Selekta Statistika Neural Network*. BP Universitas Diponegoro. Semarang.