

PENJADWALAN PERKULIAHAN OTOMATIS BERBASIS *FUZZY LOGIC* DAN *GENETIC ALGORITHM* PADA UNIVERSITAS DIAN NUSWANTORO

Edi Sugiarto¹, Sri Winarno², Amiq Fahmi³

^{1,3}Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Nakula 1 No 5-11 Semarang 50131 (024) 3569196

²Faculty of Information Science and Technology, Multimedia University

Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia

Email: edi.sugiarto@dsn.dinus.ac.id¹, fattahmg@gmail.com², amfadns@gmail.com³

Abstrak

Penjadwalan kuliah pada perguruan tinggi merupakan proses kegiatan yang penting dalam kegiatan operasional sebelum proses belajar mengajar dimulai, proses penjadwalan ini juga merupakan persoalan yang rumit dan memerlukan ketelitian karena hasil dari proses penjadwalan perkuliahan yang kurang baik akan berdampak pada penggunaan sumber daya yang tidak optimal seperti alokasi ruang, waktu belajar, beban pengajar. Penelitian ini membahas tentang perancangan dan pengembangan sistem penjadwalan otomatis dengan menggunakan algoritma genetika yang dioptimasi dengan fuzzy logic untuk mempercepat proses penjadwalan. Algoritma genetika diusulkan pada penelitian ini karena telah terbukti dapat digunakan untuk melakukan komputasi masalah penjadwalan meskipun dalam penelitian yang pernah dilakukan menyebutkan bahwa algoritma genetika ini masih memiliki kelemahan dalam hal konvergensi yang terlalu cepat sehingga tidak ada jaminan bahwa solusi yang dihasilkan adalah solusi yang optimal. fuzzy logic dipilih untuk dikombinasikan dengan algoritma genetika guna mendapatkan performance, aturan fuzzy logic yang diterapkan pada operator genetika seperti crossover dan mutasi membuat algoritma genetika yang diterapkan menjadi lebih praktis. Dari eksperimen yang telah dilakukan dengan melakukan simulasi penjadwalan, dari hasil eksperimen tersebut dibuktikan bahwa dengan setting parameter genetika 40 kromosom, crossover rate 10% dan mutation rate sebesar 10% dihasilkan global optimum yang rata-rata dicapai pada generasi ke 18 dengan waktu komputasi sebanyak 267 detik atau 4.45 menit. Sehingga dengan menambahkan fuzzy logic pada algoritma genetika terbukti bahwa proses penjadwalan perkuliahan secara otomatis menjadi lebih optimal.

Kata Kunci : Penjadwalan, Algoritma Genetika, Fuzzy Logic

Abstract

Scheduling class at university is an important activity in operations prior to the learning process begin, the scheduling process is also an issue that is complicated and requires precision because poor process of scheduling could result on non optimal use of resource such of class allocation, class time, and lecturer load. This study discusses the design and development of Automated scheduling system using a genetic algorithm that is optimized with fuzzy logic to accelerate the process of scheduling. Genetic algorithms proposed in this study because it has proven it can be used to perform computing scheduling problems despite the studies that have been conducted to mention that genetic algorithms it has limitations in terms of convergence that is too fast so there is no guarantee that the resulting solution is an optimal solution. Fuzzy logic chosen to be combined with genetic algorithm in order to gain performance, the rules of fuzzy logic applied to the genetic operators such as crossover and mutation makes genetic algorithm is applied to more practical. From the experiments that have been performed by simulating scheduling, the results of the experiment proved that by setting the parameter genetics 40 chromosomes, crossover rate of 10% and a mutation rate of 10% is produced globally optimum average achieved in generation 18 with the computational time as much as 267 4:45 seconds or minutes. So by adding a fuzzy logic on genetic algorithm proved that the process of scheduling lectures automatically become more optimal.

Keywords : scheduling, genetic algorithm, fuzzy logic

1. PENDAHULUAN

Penjadwalan kuliah pada perguruan tinggi merupakan proses kegiatan yang penting dalam kegiatan operasional sebelum proses belajar mengajar dimulai [1], proses penjadwalan ini juga merupakan persoalan yang rumit dan memerlukan ketelitian karena hasil dari proses penjadwalan perkuliahan yang kurang baik akan berdampak pada penggunaan sumber daya yang tidak optimal seperti alokasi ruang, waktu belajar, beban pengajar. Hal ini juga dapat berdampak pada bertambahnya biaya operasional yang dikeluarkan [2][3][1]. Penelitian tentang penjadwalan kuliah telah dilakukan dan sebagian besar penelitian dibidang ini berfokus pada penjadwalan kelas dan penjadwalan dosen [3]. Kesulitan dalam proses penjadwalan dijabarkan dalam beberapa hal yang harus diatasi antarlain bagaimanakah algoritma penjadwalan ini mampu menghasilkan spesifikasi jadwal yang terbebas dari benturan penggunaan ruangan, alokasi penggunaan ruangan yang optimal dengan jeda waktu perkuliahan, waktu pengajar yang tidak berbenturan dan juga harus memperhatikan rasio beban mengajar dalam satu hari atau dalam satu minggu [4].

Universitas Dian Nuswantoro didirikan pada 30 agustus 2001 berdasarkan SK Mendiknas RI No. 169/D/O/2001, yang merupakan penggabungan dari Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK), Sekolah Tinggi Ilmu Ekonomi (STIE), Sekolah Tinggi Bahasa Asing (STIBA) dan Sekolah Tinggi Kesehatan (STKES). Dalam penggabungan tersebut maka STMIK menjadi Fakultas Ilmu Komputer, STIE menjadi Fakultas Ekonomi, STBA menjadi Fakultas Bahasa dan Sastra, STKES menjadi Fakultas Kesehatan

dan bertambah satu fakultas baru yakni Fakultas Teknik[5]. Fakultas Ilmu Komputer memiliki 7 program studi yakni Teknik Informatika-S1, Sistem Informasi-S1, Desain Komunikasi Visual-S1, Ilmu Komunikasi-S1, Teknik Informatika-D3, Manajemen Informatika-D3, dan Broadcasting-D3, serta merupakan salah satu fakultas terbesar pada Universitas Dian Nuswantoro dari segi jumlah mahasiswa aktif per semester, dengan semakin berkembangnya fakultas maka diperlukan dukungan dari sisi sistem informasi yang dapat mengakomodasi transaksi operasional fakultas, proses penjadwalan perkuliahan pada fakultas ilmu komputer di tangani oleh pihak tata usaha fakultas dengan koordinasi ketua program studi mengenai penawaran matakuliah pada tiap awal semester, beberapa permasalahan yang ditimbulkan dari sisi teknis penjadwalan matakuliah antarlain sistem penjadwalan saat ini yang semi otomatis sehingga masih tergantung dengan peran operator dalam proses penjadwalan, meskipun aplikasi telah dilengkapi dengan validasi seperti validasi penggunaan waktu kuliah yang sama, benturan ruangan, namun karena keterbatasan tenaga kesalahan entri data sering terjadi, dan proses memperbaiki kesalahan tersebut dan lain sebagainya membuat waktu penjadwalan menjadi lebih lama.

Algoritma genetika merupakan teknik pencarian stokastik yang berkerja berdasarkan mekanisme seleksi alami dan genetika alami. Algoritma genetika ditemukan oleh John H. Holland dari university of Michigan pada tahun 1960 dan merupakan algoritma optimasi yang tidak berdasarkan pada matematika namun berdasarkan fenomena alam yang dalam penelusurannya mencari titik optimal berdasarkan ide yang ada pada

genetika dan teori Darwin *survival of the fittest* [6][7][8]. Algoritma genetika telah diujikan oleh Oluwasefunmi dari universitas Chinese Academy of Science, Beijing, China dalam penelitiannya untuk penjadwalan ujian pada University of Agriculture, Abeokuta, Nigeria. Dalam penelitiannya menyimpulkan bahwa algoritma ini dapat digunakan untuk optimalisasi penjadwalan ujian meskipun tidak sepenuhnya optimal namun algoritma ini dapat memberikan hasil optimal terdekat [3]. Dalam beberapa penelitian menyebutkan algoritma genetika memiliki kelemahan karena konvergensi yang terlalu cepat sehingga tidak ada jaminan solusi yang di hasilkan adalah solusi yang optimal[9].

Fuzzy logic merupakan bagian atau salah satu metode dalam kecerdasan buatan yang diperkenalkan oleh Lotfi Zadeh pada 1965 yang berprinsip bahwa logika benar dan salah tidak dapat mengatasi masalah yang ada pada dunia nyata [9][10]. Fuzzy logic pernah digunakan oleh Huaei-Xiang Zang [11] untuk memecahkan persoalan local optimal dan lambatnya diversity populasi yang sering terjadi pada algoritma genetika. Dalam penelitian tersebut diusulkan penggunaan logika fuzzy yang digabungkan dengan algoritma genetika untuk meningkatkan *evolutionary speed* dan *population diversity* dimana selanjutnya proses crossover dan mutasi pada algoritma genetika diterapkan berdasarkan aturan yang berbasis *fuzzy logic*. Hasil penelitian tersebut menunjukkan bahwa dengan menambahkan aturan fuzzy logic maka algoritma genetika yang digunakan lebih praktis dan efektif

Berdasarkan paparan diatas maka diusulkan penelitian tentang "Rancang Bangun Sistem Penjadwalan

Perkuliahan Otomatis Berbasis *Fuzzy Logic* dan *Genetic Algorithm* Pada Universitas Dian Nuswantoro" dan diharapkan hasil dari penelitian ini berupa sebuah perangkat lunak yang dapat digunakan untuk melakukan penjadwalan perkuliahan secara otomatis sehingga lebih efektif dan efisien.

Beberapa penelitian terkait tentang penjadwalan perkuliahan yang telah dilakukan antara lain sebagai berikut. Enze Yu dan Ki Seok Sung dari Seoul National University Korea melakukan penelitian tentang penjadwalan kuliah menggunakan sector based genetic algorithm, dimana metode tersebut merupakan kombinasi algoritma genetika dan *sector based partially mapped crossover* (SB-PMX) dimana metode ini digunakan untuk aturan crossover dan mutasi [12]. Eksperimen dan analisa dilakukan dengan jumlah populasi 25, 50 dan 100, dengan crossover rate dan mutation rate sebesar 0.6 dan 0.07. dari eksperimen yang dilakukan terlihat bahwa dengan populasi 100 algoritma Sector Based – Genetic Algorithm ini memiliki hasil yang terbaik.

Oluwasefunmi T. Arogundade dari Chinese Academy of Science Beijing, China [3] melakukan penelitian mengenai penjadwalan ujian pada Universitas Agriculture Abeokuta, Nigeria berbasis *algoritma genetika*. Penelitian tersebut ditujukan untuk memecahkan persoalan yang dihadapi dalam proses penjadwalan ujian pada universitas tersebut seperti menentukan slot waktu ujian yang tepat untuk para murid, menentukan nomor kursi untuk ujian para murid, dan mengoptimalkan waktu proses pembuatan ujian. Dalam penelitian tersebut ditentukan untuk hard constrain dan soft constrain yang

dapat dilalui dalam proses penjadwalan antaralain : mahasiswa tidak boleh memiliki dua waktu ujian yang sama, untuk tiap slot waktu mahasiswa yang melakukan ujian tidak boleh melebihi kapasitas tempat duduk maksimum yang disediakan. dalam penelitiannya algoritma genetika merupakan algoritma yang digunakan untuk melakukan optimasi penjadwalan ujian dengan tahapan-tahapan siklus sebagai berikut : 1. Menentukan sejumlah n kromosom secara acak untuk membentuk awal populasi, 2. Menentukan nilai tiap individu pada populasi, 3. Verifikasi kriteria terminasi, 4. Memilih $n/2$ pasang dari kromosom untuk crossover, 5. Mereproduksi kromosom dengan rekombinasi dan mutasi, 6. Menghasilkan populasi baru dari kromosom tersebut yang dinamakan generasi baru, 7. Kembali ke langkah 2. Eksperimen dilakukan dengan melakukan proses penjadwalan untuk ujian sebanyak 437 ujian, dengan 8000 mahasiswa, jumlah hari ujian adalah 22 hari dan dengan 42 ruang kelas. Dari eksperimen yang dilakukan dalam penelitian ini disimpulkan bahwa algoritma genetika dapat digunakan untuk memecahkan persoalan optimasi penjadwalan meskipun tidak sepenuhnya optimal namun mampu menghasilkan solusi yang terdekat.

2. METODE

2.1 Penjadwalan

Penjadwalan merupakan semua kegiatan untuk membuat jadwal yang tepat atas pekerjaan terhadap sumber daya yang tersedia dengan constrain atau batasan-batasan yang harus dipenuhi. Sedangkan pengertian Jadwal merupakan tabel peristiwa yang diatur sesuai dengan waktu yang tepat dan rencana urutan kerja [13][14]. Berbagai

penelitian tentang penjadwalan telah dilakukan, umumnya persoalan penjadwalan muncul pada beberapa area seperti penjadwalan kerja perawat, penjadwalan olahraga, transportasi, penjadwalan pendidikan dll [15]. Pada intinya sebuah jadwal harus memenuhi keinginan dan dibuat dengan sebaik-baiknya guna meningkatkan efisiensi penggunaan sumber daya.

2.2 Penjadwalan Perkuliahan

Penjadwalan perkuliahan pada universitas merupakan bagian dari bidang penjadwalan pendidikan atau yang sering disebut University Timetabling Problem (UTP) yang bertujuan membuat perencanaan jadwal kegiatan belajar mengajar pada suatu lembaga pendidikan [15].

Beberapa aspek yang perlu diperhatikan dalam proses penjadwalan perkuliahan sebagian besar meliputi kendala yang harus dihindari (hard constraint) maupun kendala yang masih dapat terjadi sesuai dengan batas toleransi (soft constraint) :

1. *Hard Constraints*

Hard constraints dalam permasalahan penjadwalan perkuliahan diantaranya meliputi:

- Seorang dosen tidak boleh mengajar dua matakuliah dalam jam dan hari yang sama.
- Jadwal perkuliahan tidak boleh menggunakan ruang kuliah pada hari dan jam yang sama untuk dua matakuliah

2. *Soft constraints*

Sedangkan *soft constraints* dalam permasalahan penjadwalan perkuliahan diantaranya meliputi:

- Seorang dosen mengajar lebih dari 3 matakuliah dalam satu hari
- Benturan hari dan jam perkuliahan suatu matakuliah dalam kelompok jadwal yang sama
- Terdapat tiga atau lebih matakuliah dengan hari yang sama dalam satu kelompok

2.3 Fuzzy Logic

Logika fuzzy merupakan bagian dari metode dalam kecerdasan buatan yang pertama kali oleh Lofti Zadeh pada 1965 yang berprinsip bahwa logika benar dan salah tidak dapat mengatasi masalah yang ada pada dunia nyata [9][10]. Tidak seperti logika boolean, logika fuzzy memiliki logika yang kontinu. Samar (*fuzzy*) dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang bersamaan. Teori himpunan individu dapat memiliki derajat keanggotaan dengan nilai yang kontinyu, bukan hanya nol dan satu [9]. Dalam prosesnya fuzzy logic melibatkan fungsi keanggotaan, operator logika fuzzy, dan aturan berbasis jika-maka (*if-then*). Dalam membangun sistem yang berbasis pada aturan fuzzy maka akan digunakan variabel linguistik. Variabel linguistik adalah suatu interval numerik dan mempunyai nilai-nilai linguistik, yang semantiknya didefinisikan oleh fungsi keanggotaannya. Suatu sistem berbasis aturan fuzzy terdiri dari tiga komponen utama [5] sebagai berikut:

1. Fuzzification

Fuzzification berfungsi untuk mengubah masukan-masukan yang nilai kebenarannya bersifat pasti (*crisp input*) ke dalam bentuk *fuzzy input*, yang berupa

nilai linguistik yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu.

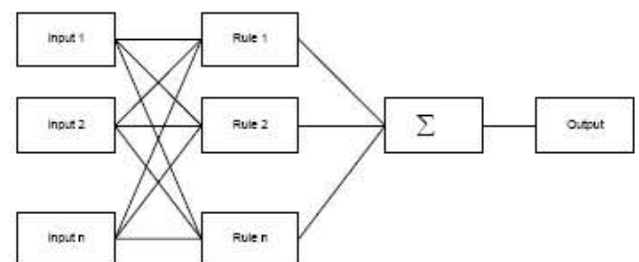
2. Inference

Inference melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Proses *inference* memperhitungkan semua aturan yang ada dalam basis pengetahuan. Hasil dari proses inference dipresentasikan oleh suatu *fuzzy set* untuk setiap variabel bebas (pada *consequent*). Derajat keanggotaan untuk setiap nilai variabel tidak bebas menyatakan ukuran komparabilitas terhadap variabel bebas (pada *antecedent*)

3. Defuzzification

Defuzzification atau penegasan berfungsi untuk mengubah *fuzzy output* menjadi *crisp value* berdasarkan fungsi keanggotaan yang telah ditentukan.

Secara garis besar proses pada fuzzy logic dapat digambarkan sbb:



Gambar 1. Proses pada fuzzy logic

2.4 Algoritma Genetika

Algoritma genetika (GA) merupakan metode algoritma pencarian yang berdasarkan pada mekanisme seleksi alam dan genetik alam yang pertama kali diperkenalkan oleh John Holland pada tahun 1975 yang dikembangkan dari bukunya yang berjudul *Adaptation in natural and artificial systems* [6].

Algoritma ini memulai serangkaian prosesnya dimulai dari membangun kromosom-kromosom yang disebut populasi, setiap kromosom terdiri dari beberapa gen yang membentuk satu kromosom, setiap gen dapat berupa nilai bit "0" atau "1" atau nilai alfabet. Selanjutnya populasi ini akan berevolusi menjadi populasi yang berbeda melalui serangkaian iterasi. Pada akhir iterasi algoritma genetika akan memiliki anggota populasi yang terbaik sebagai solusi, solusi yang terbaik ini diketahui berdasarkan suatu kondisi yang memaksimalkan kecocokannya atau kelazimannya yang disebut *fitness* [9]. Beberapa aspek penting dalam penggunaan algoritma genetika antaralain : definisi fungsi fitness, implementasi representasi genetic, dan implementasi operasi genetic.

2.4.1 Operator Algoritma Genetika

Algoritma genetik merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal tersebut diperlukan guna menghindari terjadinya konvergensi yang premature artinya solusi yang optimum tercapai sebelum waktunya, dengan katalain bahwa solusi tersebut merupaakn solusi local. Operator-operator tersebut antaralain [9] :

a. Reproduksi

Reproduksi merupakan suatu proses dimana sebuah string individu disalin kembali menjadi individu baru yang siap menjadi orang tua. Mekanisme ini menggunakan *cross over* (kawin silang) dan mutasi. Sedangkan pemilihan individu orang tua bergantung pada

fungsi fitness dari individu tersebut dalam populasinya. Kegunaan dalam pemilihan orang tua adalah dalam GA untuk memberikan kesempatan bereproduksi terhadap seluruh anggota populasi yang terbaik.

b. Cross Over

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Beberapa jenis crossover adalah:

c. Mutasi

Mutasi dalam GA merupakan proses mengubah nilai dari satu atau beberapa gen dalam satu kromosom. Proses mutasi dalam system biologi berlangsung dengan mengubah isi *llele* gen pada suatu *locus* dengan *allele* yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan *fitness* yang lebih baik. Proses mutasi perlu dilakukan dengan probabilitas yang kecil karena jika mutasi dilakukan terlalu sering akan menghasilkan individu yang lemah karena konfigurasi bit pada kromosom yang unggul akan rusak.

2.4.2 Parameter Algoritma Genetika

GA memerlukan 4 parameter [6] diantaranya :

1. Probabilitas Persilangan

Menunjukkan kemungkinan *crossover* terjadi antara 2 kromosom. Jika tidak terjadi *crossover* maka keturunannya akan sama persis dengan

kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika probabilitas *crossover* 100% maka semua keturunannya dihasilkan dari *crossover*. *Crossover* dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik

2. Probabilitas Mutasi

Menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah kromosom. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah *crossover* tidak berubah. Jika terjadi mutasi bagian kromosom akan berubah. Jika probabilitas 100%, semua kromosom dimutasi. Jika probabilitasnya 0%, tidak ada yang mengalami mutasi.

3. Jumlah Individu

Menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetik akan mempunyai sedikit variasi kemungkinan untuk melakukan *crossover* antara orangtua karena hanya sebagian kecil dari *search space* yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetik akan berjalan lambat.

4. Jumlah Populasi

Menentukan jumlah populasi atau banyaknya generasi yang dihasilkan, digunakan sebagai batas akhir proses seleksi, persilangan dan mutasi.

2.5 Penentuan Masalah

permasalahan yang sering terjadi adalah berkaitan dengan kendala yang harus dihindari dalam penjadwalan, kendala

ini dibagi kedalam dua hal yakni kendala yang harus dihindari (hard constraint) dan kendala yang masih bisa terjadi selama batas toleransi (soft constraint).

2.5.1 Hard Constraint

Kendala yang harus dihindari dalam proses penjadwalan perkuliahan antaralain:

1. Tidak boleh ada lebih dari satu kelompok menggunakan ruangan dalam waktu yang sama
2. Kelas pagi tidak boleh di plot ke jadwal sore dan juga sebaliknya
3. Kelas praktikum tidak boleh di plot ke kelas teori dan juga sebaliknya
4. Kelas diplot jadwal sesuai dengan jumlah sksnya.
5. Kelas dengan sks sebanyak 4 sks boleh diplot pada hari dan jam yang sama untuk jadwal1 dan jadwal2-nya

2.5.2 Soft Constraint

Kendala yang masih boleh terjadi namun selama masih dalam batas toleransi pada penjadwalan perkuliahan antaralain meliputi:

- Benturan hari dan jam perkuliahan suatu matakuliah dalam kelompok jadwal yang sama
- Terdapat tiga atau lebih matakuliah dengan hari yang sama dalam satu kelompok

Kelas dengan sks sebanyak 4 sks boleh diplot pada hari yang sama.

2.6. Pengumpulan Data

Data yang diperoleh berupa data sekunder yang merupakan dokumen penawaran jadwal perkuliahan yang dikumpulkan dari program studi sebagai inputan untuk penjadwalan perkuliahan. Dokumen penawaran tersebut berisi mengenai detail matakuliah yang

ditawarkan pada tahun ajaran tersebut beserta dengan jumlah kelas yang ditawarkan baik kelas pagi maupun malam.

2.7. Pengolahan Data Awal

Dari dokumen penawaran yang telah dikumpulkan selanjutnya dilakukan pengolahan data awal sebelum proses penjadwalan, data tersebut merupakan informasi mengenai matakuliah, jumlah ditawarkan kelas pagi, dan jumlah ditawarkan kelas sore. Sebagai contoh : matakuliah Dasar Pemrograman pada program studi Teknik Informatika ditawarkan pada semester tersebut dengan jumlah kelompok yakni 12 kelas pagi dan 3 kelas sore maka pencatatannya sbb:

Tabel 1: Contoh mata kuliah

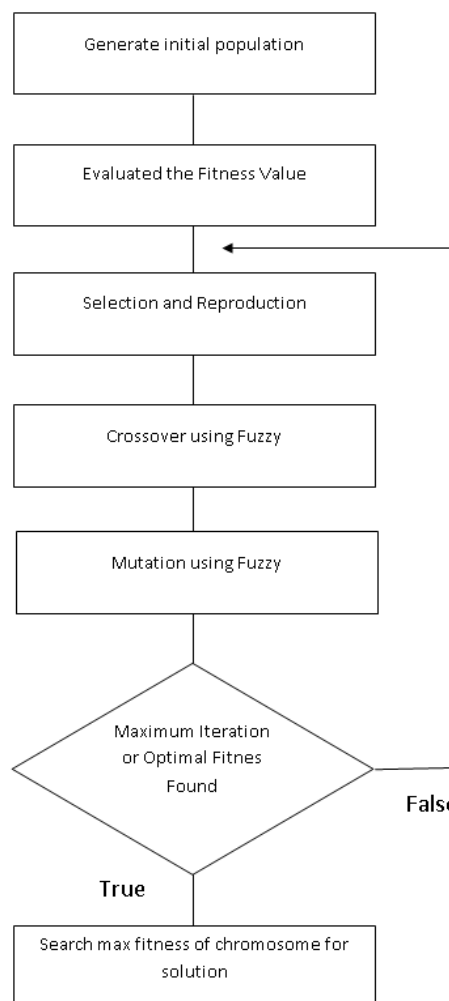
Kode_MK	Nama_MK	SK S	Jml_Pagi	Jml_Sore
A11.44106	Dasar Pemrograman	4	12	3

Dari data awal tersebut digunakan sebagai acuan untuk pembentukan jumlah kelompok yang nantinya menjadi gen untuk tiap kromosom pada proses penjadwalan dengan algoritma genetika. Sehingga pada implementasinya tiap kromosom akan tersusun dari gen-gen yang menyatakan jadwal perkuliahan sehingga satu kromosom merupakan satu solusi permasalahan penjadwalan.

2.8. Penerapan Algoritma Genetika dan Fuzzy Logic untuk Penjadwalan

Proses penjadwalan pada penelitian ini dibentuk dengan menggunakan algoritma genetika yang dioptimasi dengan menambahkan aturan fuzzy logic yang ditambahkan pada operator genetika yakni pada crossover dan mutasi. Hal ini dilakukan guna mempercepat proses evolusi untuk mencapai global optimum. Proses

penjadwalan dengan algoritma genetika dan fuzzy logic digambarkan dalam diagram berikut:



Gambar 2. Proses Penjadwalan dengan Algoritma Genetika

2.8.1 Representasi Individu

Dalam penelitian ini kromosom merupakan satu jadwal perkuliahan secara utuh untuk semua pertemuan kuliah. Tiap kromosom terdiri dari sejumlah gen dimana tiap gen menyimpan informasi mengenai matakuliah yang dijadwalkan sedangkan urutan gen menginformasikan jadwal perkuliahan. Sebagai contoh jika dalam satu semester terdapat 950 kelompok maka tiap kromosom berisi 950 gen dan nilai gen merupakan nilai numerik yang

merupakan ID jadwal dari matakuliah yang dijadwalkan yang telah dibentuk pada dataset pada tahap pengolahan data awal.

Selanjutnya dari data-data tersebut representasi ke dalam kromosom dapat dilihat pada tabel berikut:

Tabel 2: Representasi kromosom untuk masalah penjadwalan (Ruang ke 1)

D.4.01							
	SN	SL	RB	KM	JM	SB	MG
07.00-08.40	23	45	2	31	62		
08.40-10.20	56	10	123	92	83		
10.20-12.00	100	17	77	185	63		
12.30-14.10	34	28	190	167	65		
14.10-16.20	22	200	16	21	189		
16.20-18.00	26	67	27	164	57		
18.00-20.10	300	214	86	86	301		
20.10-21.50	150	145	20	287	147		

Tabel 3: Representasi kromosom untuk masalah penjadwalan (Ruang ke 29)

D.5.15							
	SN	SL	RB	KM	JM	SB	MG
07.00-09.30	38	19	678	911	906		
09.30-12.00	165	390	843	675	211		
12.30-15.00	168	789	92	419	73		
15.30-18.00	450	267	72	420	70		
18.30-21.00	367	902	433	562	200		

Sehingga representasi kromosomnya sebagai berikut :

23	56	100	34	22	26	300	200
Gen 1	Gen 2	Gen 3							Gen 950

2.8.2 Fungsi Fitness

Fungsi fitness merupakan fungsi tujuan atau dapat dikatakan sebagai fungsi yang menyatakan kekuatan dari suatu kromosom. Banyak cara untuk membangun fungsi fitness. Untuk mendapatkan fungsi fitness yang baik kita perlu memahami batasan-batasan yang diberikan sehingga dapat menentukan prioritas batasan yang ada. Dalam kasus penjadwalan batasan tersebut dapat ditentukan dari *hard constrain* dan *soft constraint* kemudian kita dapat memberikan penalty untuk setiap pelanggaran constraint. Dalam penelitian ini ditentukan nilai pinalti tiap pelanggaran constraint pada tabel berikut:

Tabel 4: Constraint dan pinalti untuk menentukan nilai fitness

No	Batasan	Pinalti
1	Tidak boleh ada lebih dari satu kelompok menggunakan ruangan dalam waktu yang sama	1000
2	Kelas pagi tidak boleh di plot ke jadwal sore dan juga sebaliknya	1000
3	Kelas praktikum tidak boleh di plot ke kelas teori dan juga sebaliknya	1000
4	Kelas dengan sks sebanyak 4 sks boleh diplot pada hari dan jam yang sama untuk jadwal1 dan jadwal2-nya	1000
5	Benturan hari dan jam perkuliahan suatu matakuliah dalam kelompok jadwal yang sama	50
6	Terdapat tiga atau lebih matakuliah dengan hari yang sama dalam satu kelompok	50
7	Kelas dengan sks sebanyak 4 sks boleh diplot pada hari yang sama	50

Dengan demikian pemberian fitnessnya dapat dilakukan ke dalam persamaan berikut:

$$f = \frac{1}{1 + \sum_i^n J_i P_i} \tag{1}$$

Dimana

- J : Jumlah Pelanggaran
- P : Nilai Pinalty

3.5.3 Proses Penjadwalan dengan Algoritma Genetika

Proses penjadwalan dengan algoritma genetika dilakukan dengan beberapa tahapan yakni dimulai pada tahap : membuat populasi baru, evaluasi nilai fitness, seleksi dan reproduksi, crossover atau kawin silang dan mutasi. kemudian tahapan dilanjutkan dengan evaluasi nilai fitness untuk semua kromosom, jika nilai fitness sudah sesuai dengan kriteria yang ditentukan atau jumlah iterasi telah mencapai maksimal generasi proses perulangan akan dihentikan dan kromosom yang memiliki nilai fitness terbaik akan dijadikan sebagai solusi penjadwalan.

a. Membuat populasi baru

Dalam penelitian ini representasi individu tiap kromosom telah ditentukan yakni bahwa tiap kromosom akan terbentuk dari kumpulan gen-gen. tiap gen pada kromosome merepresentasikan waktu/jadwal dan matakuliah yang dijadwalkan sehingga untuk jumlah gen ditentukan berdasarkan jumlah kelompok atau kelas yang dijadwalkan.

$$\bar{Q}_i = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_{15} \dots \mu_m) \quad (2)$$

Dimana \bar{Q} adalah kromosom dan μ merupakan nilai acak yang dibangkitkan sebagai nilai awal. Nilai acak pada tiap gen merupakan nilai acak bertipe numeric yang merupakan ID dari matakuliah yang dijadwalkan. Pseudo code dari proses ini digambarkan sbb:

Deskripsi

jmlChromosome \leftarrow Jumlah Kromosom

jmlGen \leftarrow jumlah gen

Implementasi

For i \leftarrow 1 to jmlChromosome

 For j \leftarrow 1 to jmlGen

 angkaAcak \leftarrow random(jmlGen)

 Chromosome(i,j) \leftarrow angkaAcak

 End for j

End for i

b. Seleksi dan Reproduksi

Proses seleksi adalah proses yang dilakukan untuk memilih kromosom-kromosom terbaik sebagai calon induk dari populasi, selanjutnya kromosom terbaik tersebut akan dipertahankan dan kromosom dengan fitness yang buruk akan dihilangkan dan diganti dengan kromosom baru yang memiliki nilai fitness yang lebih baik. Langkah seleksi kromosom dimulai dari tahapan-tahapan berikut:

- Menentukan nilai total fitness
Total fitness semua kromosom perlu dihitung untuk mencari nilai probabilitas kromosom. Perhitungan total fitness dilakukan dengan menjumlahkan nilai fitness terhadap semua kromosom seperti pada persamaan berikut:

$$F_{tot} = \sum_i^m F(\bar{Q}_i) \quad (3)$$

Dimana F merupakan fungsi fitness untuk tiap kromosom

- Mencari nilai probabilitas
Selanjutnya tentukan nilai probabilitas untuk tiap kromosome dengan persamaan berikut:

$$P = \frac{F(\bar{Q}_i)}{F_{tot}} \quad (4)$$

Dimana P merupakan probabilitas tiap kromosom.

- Memilih kromosom terbaik dengan Roulette Wheel
Proses seleksi dengan Roulette Wheel akan menempatkan kromosom dengan nilai fitness

tertinggi memiliki peluang terpilih lebih besar. Tahapan dalam Roulette Wheel dimulai dengan langkah berikut:

- o Menentukan nilai Cumulative Probabilitas Nilai cumulative dihitung dengan menggunakan persamaan berikut:

$$C(\bar{Q}_i) = \sum_{j=1}^i P_j \quad (5)$$

Dimana C merupakan nilai cumulative dari penjumlahan nilai probabilitas tiap kromosom, sehingga setelah ditentukan nilai cumulatifnya kromosom dengan nilai probabilitas yang tertinggi (atau kromosom dengan fitness terbaik) memiliki peluang terpilih lebih besar.

- o Setelah nilai cumulative terbentuk selanjutnya membangkitkan bilangan acak sebanyak jumlah kromosom untuk menentukan kromosom yang terpilih. Pseudo code pada proses ini sbb:

Implementasi

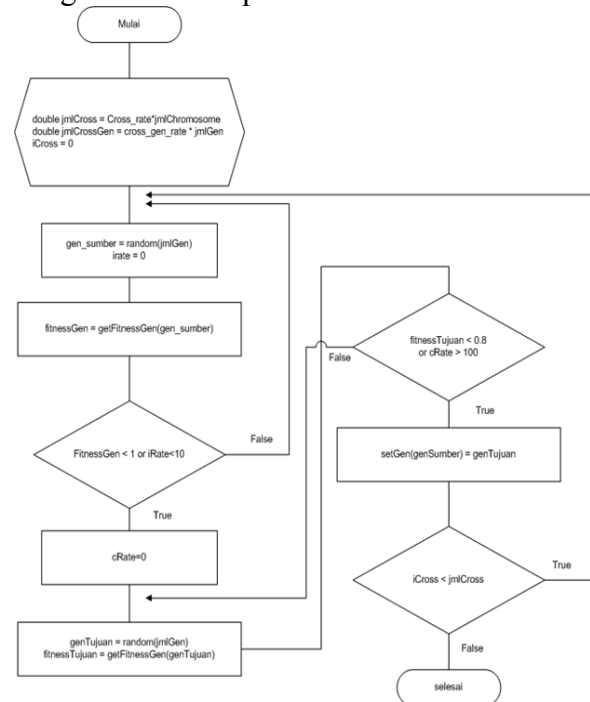
```

For I ← 1 to jmlChromosome
  R[i] ← Random(0,1)
End For
For i ← 1 to jmlChromosome
  For j ← 1 to jmlChromosome
    If (R[i] < Cumulativ[i])
      Chromosome[i] ← Chromosome[j]
    End if
  End For j
End For i
    
```

c. Crossover dengan menerapkan *Fuzzy Logic*

Setelah proses seleksi kromosom maka kromosom-kromosom dengan kemungkinan fitness terbaik terpilih, langkah selanjutnya adalah melakukan

crossover atau kawin silang. Dalam penelitian ini crossover dilakukan dengan *One-Cut Point* yakni dengan memilih satu gen secara acak pada kromosom pertama dan kemudian ditukar dengan gen pada kromosom yang lain. Dalam penelitian ini jumlah kromosom dan gen yang di crossover tergantung pada nilai parameter *crossover_rate* dan *crossover_gen_rate*, *crossover_rate* akan menentukan berapa jumlah kromosom yang akan dikawin silangkan dan *crossover_gen_rate* menentukan berapa jumlah gen pada tiap kromosom yang akan di crossover. Pada aturan crossover ditambahkan aturan fuzzy untuk menentukan apakah suatu gen diperlukan crossover atau tidak tergantung pada kondisi tertentu. Diagram alir dari proses ini sbb:

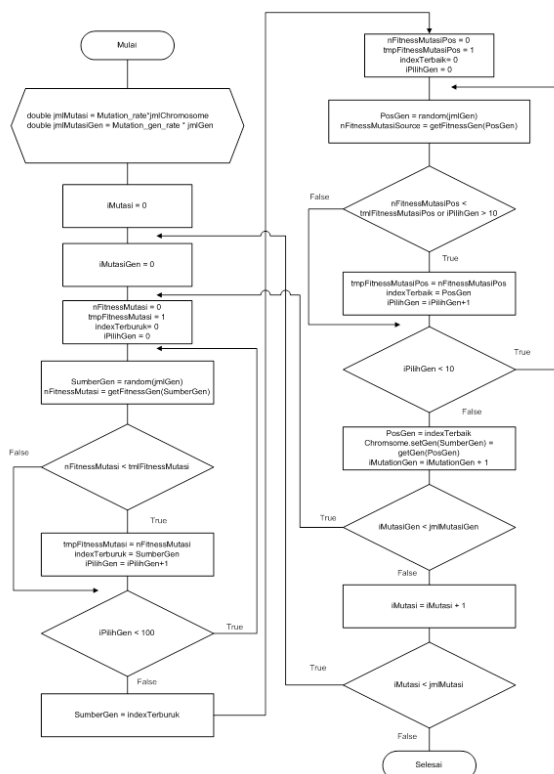


Gambar 3. Diagram alir proses Crossover dengan algoritma genetika dan fuzzy logic

d. Mutasi dengan menerapkan *Fuzzy Logic*

Mutasi dilakukan dengan memilih sembarang kromosom biasanya dilakukan secara acak dan kemudian memilih gen pada kromosom yang akan dimutasi, prosesnya adalah mengubah

nilai pada gen dengan nilai yang baru sehingga memungkinkan perubahan gen yang akan mengarah pada fitness yang lebih baik pada kromosom. Pada penelitian ini proses mutasi tergantung pada dua variable yakni *mutation_rate* dan *mutation_gen_rate*, *mutation_rate* digunakan untuk menentukan jumlah kromosom yang mengalami mutasi pada satu generasi sedangkan variable *mutation_gen_rate* digunakan untuk menentukan jumlah gen yang bermutasi pada tiap kromosom dalam satu generasi. Proses mutasi pada penelitian ini dapat dilihat pada diagram alir berikut:



Gambar 4. Diagram alir proses mutasi dengan algoritma genetika dan fuzzy logic

3. HASIL DAN PEMBAHASAN

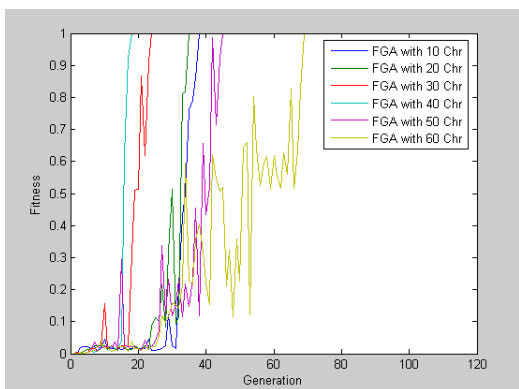
Objek penelitian pada penelitian ini adalah Fakultas Ilmu Komputer Universitas Dian Nuswantoro yang beralamat pada Jl. Nakula I No 5-11 Semarang. Penelitian dilakukan pada

fakultas tersebut yang meliputi 7 program studi yakni : Teknik Informatika - S1, Sistem Informasi - S1, Desain Komunikasi Visual – S1, Ilmu Komunikasi – S1, Manajemen Informatika – D3, Teknik Informatika – D3, dan Broadcasting – D3. Eksperimen dilakukan dengan melakukan simulasi proses penjadwalan. Proses penjadwalan dalam penelitian ini dilakukan dengan menggunakan percobaan 10 hingga 60 kromosom dengan jumlah gen tiap kromosom sebanyak 953 (sesuai dengan jumlah kelompok kuliah yang di jadwal) serta parameter crossover rate 10% dan mutation rate 10%, maksimal iterasi dalam eksperimen diatur hingga 5000 generasi. Pada eksperimen pertama percobaan dilakukan dengan menggunakan program penjadwalan yang menerapkan algoritma genetika, dan pada eksperimen kedua percobaan dilakukan menggunakan program penjadwalan yang menerapkan algoritma genetika dan dioptimasi dengan algoritma fuzzy logic. Hasil percobaan dapat dilihat pada tabel berikut:

Tabel 5: Tabel Pengukuran Proses Penjadwalan

Jumlah Kromosom	GA + Fuzzy	
	Rata-rata Jml Generasi	Total Waktu (Second)
10	38	674
20	34	642
30	23	279
40	18	267
50	44	831
60	68	1496

Grafik pengukuran dari tabel diatas dapat dilihat pada gambar berikut:



Gambar 5. Grafik Pengukuran

Dari tabel pengukuran diatas dapat dilihat bahwa solusi terbaik yang dicapai penjadwalan kuliah dengan algoritma genetika dan fuzzy yakni pada jumlah kromosom 40 dengan jumlah rata-rata generasi yakni 18 generasi dengan total waktu 267 detik atau 4.45 menit. Jika dibandingkan dengan penjadwalan yang menerapkan algoritma genetika dapat dilihat pada tabel dibawah:

Tabel 6: Setting parameter

Jumlah Kromosom	Jumlah Gen	Crossover Rate	Mutasi Rate	Maksimum Generasi
40	953	10%	10%	5000

Tabel 7: Tabel Perbandingan proses penjadwalan antara algoritma genetika standar dengan algoritma genetika dan fuzzy logic

GA		GA + Fuzzy	
Rata-rata Jml Generasi	Total Waktu (Second)	Rata-rata Jml Generasi	Total Waktu (Second)
628	2872	18	267

Hasil dari komparasi aplikasi penjadwalan yang menerapkan algoritma genetika dengan aplikasi penjadwalan yang menerapkan algoritma genetika dioptimasi dengan fuzzy logic terbukti bahwa algoritma genetika dan fuzzy logic dapat meningkatkan performance proses penjadwalan.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan dengan menerapkan algoritma genetika yang dioptimasi dengan *fuzzy logic* untuk aplikasi penjadwalan dapat diambil kesimpulan bahwa dengan menambahkan *fuzzy logic* pada operator *crossover* dan *mutation* pada algoritma genetika maka dapat meningkatkan performance algoritma genetika tersebut, hal ini dibuktikan dengan melakukan eksperimen penjadwalan pada Fakultas Ilmu Komputer Universitas Dian Nuswantoro untuk 953 kelompok atau kelas dengan setting parameter genetika 40 kromosom, crossover rate 10% dan mutation rate sebesar 10% dihasilkan *global optimum* yang rata-rata dicapai pada generasi ke 18 dengan waktu komputasi sebanyak 267 detik atau 4.45 menit, dibandingkan dengan algoritma genetika standar dengan setting parameter yang sama didapatkan *global optimum* pada generasi ke 628 dengan waktu komputasi 2872 detik atau 47 menit. Sehingga dengan menambahkan *fuzzy logic* pada algoritma genetika proses penjadwalan perkuliahan menjadi lebih optimal.

4.2 Saran

Berdasarkan dari hasil analisa dan pembahasan pada penelitian ini, terdapat beberapa saran untuk pengembangan penelitian penjadwalan otomatis selanjutnya. Algoritma genetika merupakan algoritma pencarian berbasis populasi yang cukup optimal digunakan pada kasus penjadwalan, dan dengan penambahan fuzzy logic pada operator crossover dan mutation terbukti menjadi lebih optimal, namun algoritma genetika masih memiliki kelemahan yakni terlalu cepat mencapai konvergensi sehingga sering terjebak pada masalah *local optimum* sehingga dalam penelitian selanjutnya perlu dilakukan penelitian komparasi

antara algoritma genetika dengan algoritma yang lain untuk kasus penjadwalan. Dalam penelitian ini penjadwalan perkuliahan masih dibatasi untuk penjadwalan kelas teori sehingga pada penelitian selanjutnya aplikasi penjadwalan otomatis ini perlu dikembangkan kembali sehingga dapat digunakan untuk melakukan penjadwalan kelas praktikum.

DAFTAR PUSTAKA

- [1] Olivia-Doria and Ben Paechter (2008), "A memetic algorithm for University Course Timetabling", IEEE International Conference on Tools with Artificial Intelligence.
- [2] N. R. Sabar and M. Ayob and G. Kendall and R. Qu (2012), "A honey-bee mating optimization algorithm for educational timetabling problems", European Journal of Operational Research, Volume 216, Issue 3, pp. 533-543, 2012.
- [3] Oluwasefunmi T. Arogundade, Adio T. Akinwale, Omotoyosi M. Aweda (2010), "A Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem", International Journal of Computer Applications (0975 – 8887) volume 12-No 5, December 2010.
- [4] Barry McCollum (), "A Perspective on Bridging the Gap between Theory and Practice in University Timetabling", PATAT'06 Proceedings of the 6th international conference on Practice and theory of automated timetabling VI Pages 3-24, ISBN:3-540-77344-4 978-3-540-77344-3.
- [5] Buku Pedoman Akademik Universitas Dian Nuswantoro 2010.
- [6] Sivanandam S.N, S.N Deepa (2008), "Introduction to Genetic Algorithm", Springer-verlag Heidelberg, Berlin.
- [7] Suyanto (2010), "Algoritma Optimasi Deterministik atau Probabilistik", Graha Ilmu, Yogyakarta.
- [8] Intan Berlianty dan Miftahol Arifin, "Teknik-Teknik Optimasi Heuristik", Graha Ilmu, Yogyakarta, 2010.
- [9] Suyanto, ST, MSc, "Soft Computing : Membangun Mesin Ber-IQ Tinggi", Informatika, Bandung, 2008.
- [10] Sivanandam S.N, Sumanti (2007), "Introduction to Fuzzy Logic using MATLAB", Springer-verlag Heidelberg, Berlin.
- [11] Huai-xiang Zhang, Bo Zhang and Feng Wang (2009), "Automatic Fuzzy Rules Generation Using Fuzzy Genetic Algorithm", 2009 Sixth International Conference on Fuzzy System and Knowledge Discovery.
- [12] Enze Yu and Ki-Seok Sung (2002), "A Genetic Algorithm for a University Weekly Course Timetabling Problem", International Federation of Operational Research Society.
- [13] Antariksa Badhuri (2009), "University Time Table Scheduling using Genetic Artificial Immune Network", 2009 International Conference on Advances in Recent Technologies in Communication and Computing.
- [14] Willemen, Robertus J (2002), "School Timetable Construction : Algorithms and Complexity, Technische Universiteit Eindhoven, ISBN 90-386-1011-4.
- [15] Edmun K Burke, et all (2007), "A Graph-Based Hyper-Heuristic for Educational Timetabling Problem", European Journal of Operational Research, 176: 177-192