

Penjadwalan Sidang Tugas Akhir Prodi Ilmu Komputasi Universitas Telkom Menggunakan Metode Algoritma Genetika Adaptif dan *Fuzzy* *Relation*

Oki Virgiawan Pramudita¹, Fhira Nhita², Annisa Aditsania³

School of Computing, Telkom University
Jl. Telekomunikasi No. 01, Bandung, Indonesia

¹virgiawanoki@gmail.com

²fhiranhita@telkomuniversity.ac.id

³annisaaditsania@gmail.com

Abstract

Determine a schedule that consist of many elements are complicated. One example of scheduling problems is scheduling of final project. Lot of combinations can be formed from the elemensts of this scheduling, but not all such combinations may be the solution. The scheduling of final project have requirements such lecturers teaching schedule and college student course schedule should not be conflicting to hearing schedule, examiners recommended in accordance with the topic of thesis that are tested and adviser should not be a examiners. Considering of many consideration in this scheduling, then it takes the optimal and efficient method so it can resolve the complexity of this scheduling. Fuzzy relations and genetic algorithm including methods that can be used to overcome this problem. Schedule that produced by fuzzy relation and genetic algorithm methods in this case have an average fitness of 0.9687, and for genetic algorithm methods only have an average fitness of 0.8243.

Keywords: *Scheduling, fuzzy relation, adaptive genetic algorithm, genetic algorithm.*

Abstrak

Menentukan sebuah jadwal yang terdiri atas banyak unsur sangatlah rumit. Salah satu contoh permasalahan dari penjadwalan yaitu penjadwalan sidang. Banyak kombinasi yang bisa dibentuk dari unsur-unsur penjadwalan sidang. Namun, tidak seluruh kombinasi tersebut dapat menjadi solusi dari penjadwalan. Penjadwalan sidang tugas akhir ini memiliki beberapa syarat diantaranya jadwal mengajar dosen dan jadwal kuliah mahasiswa tidak boleh bentrok dengan jadwal sidang, dosen penguji direkomendasikan sesuai dengan topik tugas akhir yang diujikan, dan dosen pembimbing tidak bisa menjadi dosen penguji. Karena banyaknya pertimbangan dalam penjadwalan sidang tugas akhir, maka dibutuhkan metode yang optimal dan efisien sehingga dapat mengatasi rumitnya penjadwalan ini. Metode *fuzzy relation* dan algoritma genetika termasuk metode yang bisa digunakan dalam mengatasi masalah ini. Jadwal yang dihasilkan oleh metode *fuzzy relation* dan algoritma genetika adaptif pada kasus ini memiliki rata-rata *fitness* sebesar 0.9687, dan untuk metode algoritma genetika saja ini memiliki rata-rata *fitness* sebesar 0.8243.

I. PENDAHULUAN

Setiap mahasiswa pasti akan menghadapi sidang tugas akhir. Sidang tugas akhir ini merupakan sebuah tahap penentuan lulus tidaknya mahasiswa di suatu Universitas, salah satunya di Universitas Telkom. Penjadwalan sidang di suatu universitas biasanya ditangani oleh program studi dari masing – masing fakultas. Pada penentuan jadwal sidang bergantung pada ketersediaan ruang sidang, waktu jadwal kosong dari mahasiswa, dua dosen pembimbing, dan dua dosen penguji, dan keahlian dari dosen.

Ada beberapa hal yang perlu diperhatikan dalam penentuan jadwal sidang tugas akhir, diantaranya jadwal sidang tidak boleh bentrok dengan jadwal mengajar dosen penguji, dosen pembimbing, jadwal kuliah mahasiswa jika mahasiswa tersebut masih kuliah, ketersediaan tempat untuk sidang, dan dosen penguji memiliki kelompok

keahlian yang sama dengan bidang TA (Tugas Akhir) mahasiswa. Kesulitan kasus penjadwalan sidang berbanding lurus dengan jumlah mahasiswa, yang dimana jika makin banyak jumlah mahasiswa yang akan sidang, maka penyusunan jadwal akan semakin rumit. Masalah yang biasa dihadapi dari penjadwalan sidang ini yaitu bentroknya jadwal dosen penguji dengan jadwal sidang sehingga dosen penguji memiliki kemungkinan tidak dapat hadir.

Penelitian ini ditujukan untuk pengoptimalan penjadwalan sidang tugas akhir agar bisa menghemat waktu penyusunan jadwal. Dengan menggunakan metode *fuzzy relation* yang digabung dengan algoritma genetika ini diharapkan dapat meningkatkan efektifitas untuk menyelesaikan permasalahan penjadwalan sidang tugas akhir yang kompleks tersebut, yang tentunya dengan lebih cepat dan tepat, sesuai dengan apa yang diharapkan.

II. TINJAUAN PUSTAKA

Terdapat 2 metode yang digunakan pada penelitian ini, metodenya yaitu *fuzzy relation* dan algoritma genetika. Metode *fuzzy relation* digunakan untuk menentukan nilai kecocokan dosen sebagai penguji terhadap mahasiswa. Sedangkan untuk metode algoritma genetika digunakan untuk menentukan jadwal sidang yang optimal.

Pada penelitian ini terdapat 2 jenis metode algoritma genetika, yaitu algoritma genetika biasa dan algoritma genetika adaptif. Perbedaan pada kedua jenis algoritma genetika tersebut yaitu, pada algoritma genetika biasa nilai probabilitas *crossover* (pc) dan probabilitas mutasi (pm) ditentukan diawal, dan nilainya tetap untuk setiap generasi. Sedangkan algoritma genetika adaptif nilai pc dan pm nya berubah-ubah untuk setiap generasi.

Sebelumnya telah dilakukan penelitian penjadwalan menggunakan algoritma genetika, yang dilakukan oleh Sujit Kumar Jha (2014), Vinny Witary (2013), Nur Rachmat (2013), dan Inayatullah (2013). Pada penelitian tersebut algoritma genetika berhasil diterapkan pada kasus penjadwalan perkuliahan dan penjadwalan ujian dengan jumlah bentrok yang sedikit. Selain itu terdapat penelitian penjadwalan sidang menggunakan *fuzzy relation* dan *harmony search* yang dilakukan oleh Annisa Dian Pratiwi (2009), pada penelitian tersebut kedua metode berhasil diterapkan dengan baik, untuk metode *fuzzy relation* cocok untuk kasus penjadwalan sidang dimana pada kasus tersebut dibutuhkan nilai keterkaitan antara dosen terhadap mahasiswa.

Fuzzy Relation

Fuzzy relation merupakan suatu metode untuk menyatakan relasi berupa fuzzy antara dua *entity* dalam sebuah wilayah/*domain*. Relasi ini dinyatakan dalam interval [0,1] yang menyatakan derajat asosiasi dari dua *entity* tersebut [4]. *Fuzzy conditional probability relation* biasanya direpresentasikan sebagai R. *Fuzzy conditional probability relation* antara dua *fuzzy set* X dan Y bernilai antara 0 sampai 1, yang didefinisikan sebagai[5]:

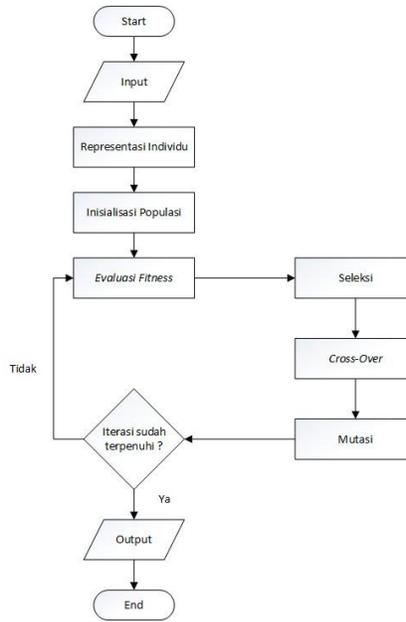
$$\mu_{Y|X}(y) = \frac{\sum_{x \in X} \mu_{X,Y}(x,y) \cdot \mu_X(x)}{\sum_{x \in X} \mu_{X,Y}(x,y)} \quad (1)$$

Dimana μ_X dan μ_Y adalah *membership function* domain D, sedangkan $\mu_{X,Y}$ merupakan derajat kemiripan Y terhadap X.

Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristic yang didasarkan pada mekanisme seleksi dan evolusi genetika secara alami [2]. Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan pada tahun 1975 di Amerika Serikat [3].

Secara umum skema dari algoritma geentika adalah seperti berikut:



Gambar 1 Skema umum algoritma genetika

Representasi Individu

Individu merupakan bentuk solusi dari permasalahan. Individu bisa dikatakan sama dengan kromosom, yang merupakan kumpulan dari gen. Dalam ‘*algoritma genetika classical*’, kromosomnya diwakili sebagai biner [7]. Misalkan suatu individu $x_1 = 7$ dan $x_2 = 5$ dikodekan menjadi kromosom 01110101. Sebaliknya kromosom 01110101 dikodekan menjadi individu $x_1 = 7$ dan $x_2 = 5$. Pada kasus ini representasi kromosom ditunjukkan pada gambar

| | | | | | | |
|-----------|-----------|-----------|---------------|--------------|--------------|--------------|
| Mahasiswa | Penguji 1 | Penguji 2 | Jadwal Sidang | Pembimbing 1 | Pembimbing 2 | Ruang Sidang |
|-----------|-----------|-----------|---------------|--------------|--------------|--------------|

Gambar 2 Representasi kromosom

Nilai Fitness

Nilai *fitness* merupakan ukuran baik tidaknya sebuah individu. Dalam dunia nyata, individu dengan nilai *fitness* yang tinggi memiliki potensi bertahan hidup yang lebih tinggi, dibandingkan dengan individu yang memiliki nilai *fitness* yang rendah [9]. Rumus *fitness* secara umum yaitu:

$$\frac{1}{1 + a \cdot x} \tag{2}$$

Dimana a adalah bilangan untuk mencegah pembilang dibagi dengan bilangan nol. Nilai x disini sebagai nilai *pinalti*. *Pinalti* merupakan nilai pelanggaran dari kendala pada suatu kromosom. Biasanya nilai *pinalti* ini sudah ditentukan. Untuk rumus *fitness* pada algoritma genetika yang digabung dengan *fuzzy relation* adalah:

$$\frac{1}{1 + a \cdot x} \tag{3}$$

Dimana P adalah nilai pinalti yang telah ditentukan, dan R adalah nilai rata-rata relasi dari kedua penguji. Untuk ketentuan nilai pinalti pada kasus ini dapat dilihat pada tabel

Tabel 1 Ketentuan pinalti

| Jenis Pelanggaran | Nilai Pinalti |
|---|---------------|
| Jadwal sidang & jadwal penguji 1 bentrok | P = 6 |
| Jadwal sidang & jadwal penguji 2 bentrok | P = 6 |
| Jadwal sidang & jadwal mahasiswa bentrok | P = 3 |
| Jadwal sidang & jadwal pembimbing 1 bentrok | P = 6 |
| Jadwal sidang & jadwal pembimbing 1 bentrok | P = 6 |
| Penguji 1 = Penguji 2 | P = 10, R = 0 |

Sedangkan untuk nilai *fitness* pada algoritma genetika saja yaitu:

$$(\quad) \quad (4)$$

Elitisme

Proses elitisme ini tidak wajib dalam algoritma genetika, yang dimana proses ini hanya sebagai tambahan saja. Elitisme ini menyalin individu terbaik agar tidak hilang selama proses *crossover* dan mutasi dan bisa ada pada generasi selanjutnya. Tujuan elitisme ini agar dapat semakin cepat ditemukannya solusi dengan nilai *Fitness* yang baik.

Seleksi Orang Tua

Pemilihan kromosom sebagai calon orang tua dilakukan secara proporsional berdasarkan nilai-nilai *fitness*-nya [10]. Seleksi ini bertujuan untuk memperoleh calon orang tua yang baik, karena calon orang tua yang baik akan menghasilkan anak yang baik.

Rekombinasi

Setelah kromosom – kromosom terpilih sebagai orang tua dilakukanlah rekombinasi untuk menghasilkan anak. Istilah rekombinasi ini bisa disebut juga dengan *crossover*. Rekombinasi ini memiliki probabilitas sehingga tidak semua orang tua melalui proses ini. probabilitas ini dinyatakan dengan variabel *pc* atau probabilitas *crossover* yang telah ditentukan. Terdapat 3 jenis rekombinasi yang biasa digunakan, yaitu rekombinasi satu titik (*1-point crossover*), rekombinasi banyak titik (*N-point crossover*), dan rekombinasi seragam (*uniform crossover*) [10].

Mutasi

Proses mutasi ini mengubah nilai gen dari kromosom. Mutasi ini terjadi secara acak pada setiap gen dan bersifat bebas [10]. Dengan kata lain tidak mempengaruhi gen-gen yang lain. Seperti proses rekombinasi, proses mutasi ini terdapat probabilitas yang dinyatakan dengan *pm* atau probabilitas mutasi yang telah ditentukan. Jika terdapat kromosom yang nilainya kurang dari *pm* yang telah ditentukan, maka dilakukanlah proses mutasi.

Seleksi Survivor

Pada seleksi survivor ini dilakukan penggantian kromosom lama dengan kromosom yang baru yang dihasilkan dari proses rekombinasi dan mutasi. Terdapat 2 macam seleksi survivor, yaitu *steady state* dan *general replacement*.

Algoritma Genetika Adaptif

Pada umumnya algoritma genetika memiliki nilai probabilitas *crossover* dan probabilitas mutasi yang tetap untuk setiap generasinya. Secara umum nilai probabilitas untuk *crossover* yaitu sebesar 0.5 – 0.9 dan untuk nilai

probabilitas mutasinya sebesar 0.001 – 0.2, atau menggunakan rumus sampai [10], dimana adalah ukuran populasi dan adalah panjang kromosom (jumlah gen) [10]. Namun pada algoritma genetika secara adaptif ini nilai dari masing – masing probabilitas tersebut berubah-ubah sesuai dengan keadaan dan kebutuhan.

Untuk rumus dari algoritma genetika adaptif-nya sebagai berikut:

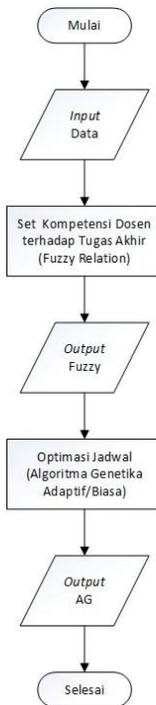
$$\{ \text{-----} \} \tag{5}$$

$$\{ \text{-----} \} \tag{6}$$

Dimana F_{max} adalah *fitness* maksimum pada populasi, F' adalah *fitness* terbesar pada calon individu *crossover*, F_{ave} adalah rata-rata *fitness* pada populasi dan F adalah *fitness* dari calon individu mutasi.. Untuk , , dan biasanya bernilai antara 0 – 1 [12]. Pada kasus ini dan nilainya sebesar 1 dan untuk dan nilainya sebesar 0.5 [12]..

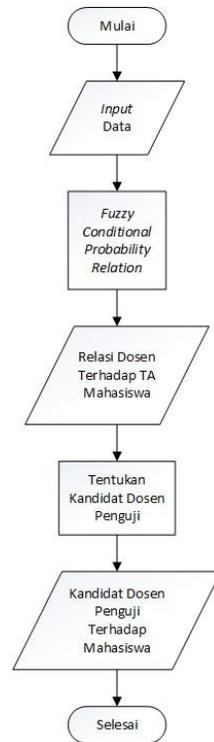
III. DESAIN SISTEM

Dalam penelitian ini, metode *fuzzy relation* dan algoritma genetika (AG) maupun algoritma genetika adaptif (AGA) digunakan untuk mngoptimasi jadwal sidang dari Prodi Ilmu Komputasi Fakultas Informatika Universitas Telkom dan diimplementasikan menggunakan software Matlab. Untuk rancangan sistem secara umumnya dapat dilihat sebagai berikut:



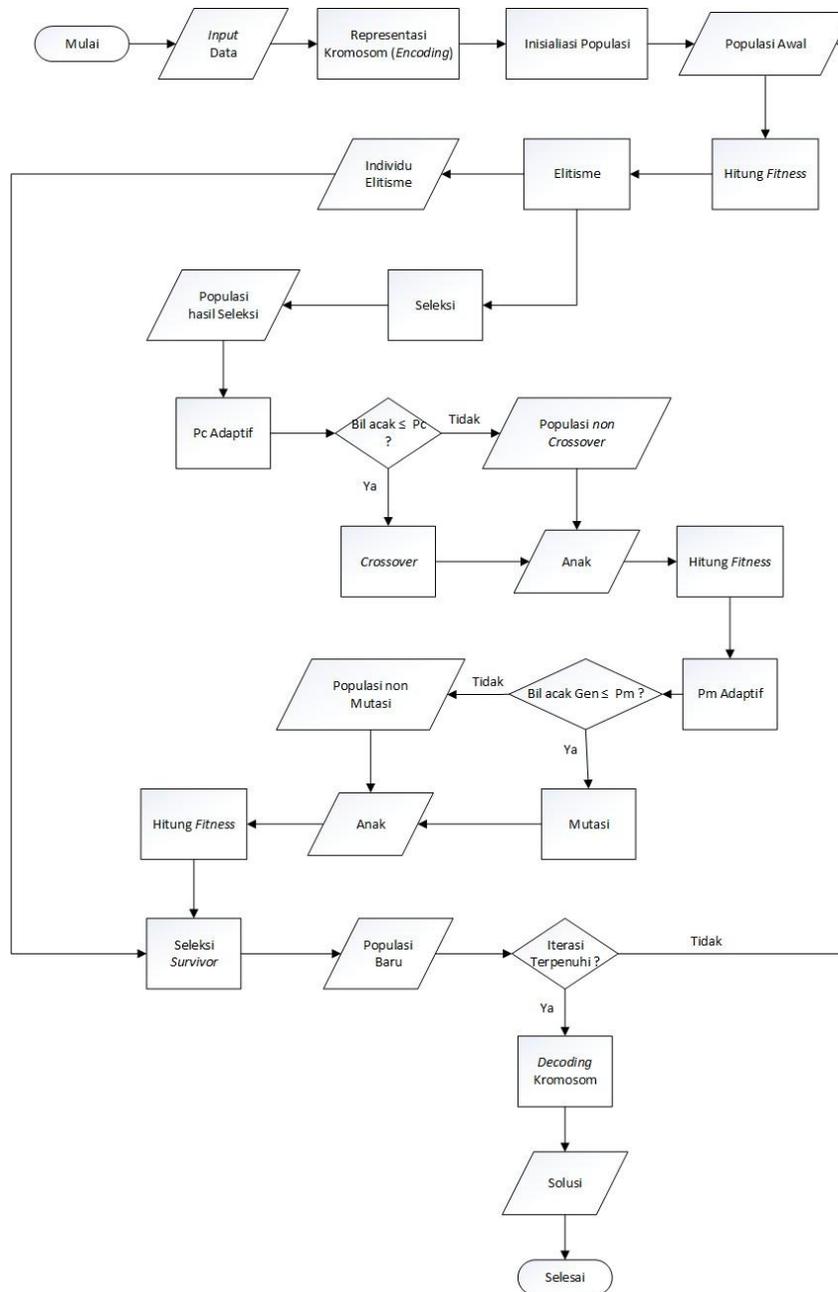
Gambar 3 Flowchart sistem secara umum

Secara umum metode algoritma *fuzzy relation* dijalankan terlebih dahulu baru algoritma genetika. Karena hasil dari *fuzzy relation* nantinya digunakan pada algoritma genetika.

Gambar 4 Flowchart *fuzzy-relation*

Pada metode *fuzzy relation* data masukannya yaitu nilai relasi antara dosen terhadap bidang, dan nilai relasi mahasiswa terhadap bidang. Bidang yang dimaksud disini yaitu bidang seputar TA mahasiswa Ilmu Komputasi. Untuk outputnya nanti berupa nilai relasi dosen terhadap mahasiswa.

Untuk metode algoritma genetika, data masukannya terdiri dari jadwal mengajar dosen, jadwal kuliah mahasiswa, jadwal sidang secara keseluruhan, data relasi dosen terhadap mahasiswa, dan info ruangan sidang.



Gambar 5 Flowchart algoritma genetika adaptif

Gambar diatas merupakan *flowchart* dari metode algoritma genetika adaptif., yang dimana nilai p_c dan p_m -nya berubah-ubah setiap generasi dan diperoleh dari perhitungan pada persamaan (2) dan (3). Untuk metode algoritma genetika yang biasa memiliki *flowchart* yang sama dengan algoritma genetika adaptif, perbedaannya hanya terletak pada nilai p_c dan p_m yang dimana nilai tersebut telah ditentukan sebelumnya dan nilainya sama untuk setiap generasi.

IV. HASIL PENGUJIAN DAN DISKUSI

Dalam pengujian ini penulis membuat tiga skenario yaitu optimasi jadwal sidang menggunakan algoritma genetika, *fuzzy relation*-algoritma genetika, dan *fuzzy relation*-algoritma genetika adaptif.

Skenario 1

Pada skenario pertama dilakukan implementasi kasus menggunakan algoritma genetika. Masing-masing parameter dilakukan percobaan sebanyak 10 kali dan diambil nilai rata-ratanya. Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 200, jumlah generasi 14, dapat dilihat pada tabel dibawah:

Tabel 2 Hasil rata-rata fitness algoritma genetika individu 200 generasi 14

| pc | pm | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-----|------|---|
| 0.5 | 0.06 | 0.8834 |
| 0.5 | 0.08 | 0.8844 |
| 0.5 | 0.1 | 0.8963 |
| 0.6 | 0.06 | 0.8824 |
| 0.6 | 0.08 | 0.8962 |
| 0.6 | 0.1 | 0.8930 |
| 0.7 | 0.06 | 0.8759 |
| 0.7 | 0.08 | 0.8943 |
| 0.7 | 0.1 | 0.8923 |
| 0.8 | 0.06 | 0.8800 |
| 0.8 | 0.08 | 0.8883 |
| 0.8 | 0.1 | 0.8951 |
| 0.9 | 0.06 | 0.8823 |
| 0.9 | 0.08 | 0.8852 |
| 0.9 | 0.1 | 0.8958 |

Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 400 dan jumlah generasi sebanyak 7, dapat dilihat pada tabel dibawah:

Tabel 3 Hasil rata-rata fitness algoritma genetika individu 400 generasi 7

| Pc | Pm | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-----|------|---|
| 0.5 | 0.06 | 0.9222 |
| 0.5 | 0.08 | 0.9227 |
| 0.5 | 0.1 | 0.9256 |
| 0.6 | 0.06 | 0.9129 |
| 0.6 | 0.08 | 0.9223 |
| 0.6 | 0.1 | 0.9277 |

| | | |
|-----|------|--------|
| 0.7 | 0.06 | 0.9192 |
| 0.7 | 0.08 | 0.9251 |
| 0.7 | 0.1 | 0.9312 |
| 0.8 | 0.06 | 0.9186 |
| 0.8 | 0.08 | 0.9216 |
| 0.8 | 0.1 | 0.9318 |
| 0.9 | 0.06 | 0.9190 |
| 0.9 | 0.08 | 0.9195 |
| 0.9 | 0.1 | 0.9333 |

Contoh hasil jadwal dari salah satu percobaan diatas dengan jumlah individu sebanyak 400, 7 generasi, nilai probabilitas *crossover* sebesar 0.7 dan probabilitas mutasi sebesar 0.1 adalah sebagai berikut:

Tabel 4 Salah satu contoh jadwal algoritma genetika

| Mahasiswa | Penguji 1 | Penguji 2 | Jadwal | Pembimbing 1 |
|--------------------|-----------|-----------|---------------------|--------------|
| Oki Virgiawan P | FTR | RFU | Sel 08:30 - 10:10 | VIR |
| Choiryaldi Setya P | ADW | KMM | Rabu 10:30 - 12:10 | VIR |
| Faishal Mandala P | IND | ZKA | Jumat 08:30 - 10:10 | ADW |
| Lucky Deka P | YLS | DLW | Jumat 12:30 - 14:10 | ADW |
| | ... | ... | ... | ... |
| Maharauna Lozanno | DLW | KMM | Sel 12:30 - 14:10 | ERW |

| Pembimbing 2 | Ruang | Fitness Relasi | Penalti | Relasi |
|--------------|-------|----------------|---------|---------|
| TSA | 1 | 0.791666667 | 0 | 0.5833 |
| TSA | 1 | 0.85 | 0 | 0.7 |
| ANQ | 1 | 0.730769231 | 0 | 0.4615 |
| ANQ | 3 | 0.971153846 | 0 | 0.9423 |
| ... | ... | ... | ... | ... |
| YLS | 1 | 0.828125 | 0 | 0.65625 |

Skenario 2

Pada skenario kedua dilakukan implementasi kasus menggunakan *fuzzy relation*-algoritma genetika. Untuk implementasi algoritma *fuzzy relation* data masukannya terdiri dari relasi dosen terhadap bidang dan relasi TA mahasiswa terhadap bidang. Berikut beberapa hasilnya:

Tabel 5 Contoh hasil fuzzy relation

| DOSEN/MAHASISWA | Choiryaldi Setya P | I Gede Komang | Andrian Yoga P |
|-----------------|--------------------|---------------|----------------|
| ADW | 0.666666667 | 0.2 | 0.416666667 |
| YLS | 0.833333333 | 0.766666667 | 0.666666667 |
| ANQ | 0.333333333 | 0.933333333 | 0.125 |

Untuk algoritma genetika masing-masing parameter dilakukan percobaan sebanyak 10 kali dan diambil nilai rata-ratanya. Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 200, jumlah generasi 14, dapat dilihat pada tabel dibawah:

Tabel 6 Hasil rata-rata fitness fuzzy relation-algoritma genetika individu 200 generasi 14

| pc | pm | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-----|------|---|
| 0.5 | 0.06 | 0.9707 |
| 0.5 | 0.08 | 0.9751 |
| 0.5 | 0.1 | 0.9730 |
| 0.6 | 0.06 | 0.9661 |
| 0.6 | 0.08 | 0.9686 |
| 0.6 | 0.1 | 0.9723 |
| 0.7 | 0.06 | 0.9636 |
| 0.7 | 0.08 | 0.9659 |
| 0.7 | 0.1 | 0.9773 |
| 0.8 | 0.06 | 0.9617 |
| 0.8 | 0.08 | 0.9732 |
| 0.8 | 0.1 | 0.9689 |
| 0.9 | 0.06 | 0.9681 |
| 0.9 | 0.08 | 0.9762 |
| 0.9 | 0.1 | 0.9712 |

Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 400 dan jumlah generasi sebanyak 7, dapat dilihat pada tabel dibawah:

Tabel 7 Hasil rata-rata fitness fuzzy relation-algoritma genetika individu 400 generasi 7

| pc | pm | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-----|------|---|
| 0.5 | 0.06 | 0.9738 |
| 0.5 | 0.08 | 0.9753 |
| 0.5 | 0.1 | 0.9749 |
| 0.6 | 0.06 | 0.9704 |
| 0.6 | 0.08 | 0.9765 |
| 0.6 | 0.1 | 0.9784 |
| 0.7 | 0.06 | 0.9761 |
| 0.7 | 0.08 | 0.9752 |
| 0.7 | 0.1 | 0.9762 |
| 0.8 | 0.06 | 0.9730 |
| 0.8 | 0.08 | 0.9732 |
| 0.8 | 0.1 | 0.9770 |
| 0.9 | 0.06 | 0.9732 |
| 0.9 | 0.08 | 0.9762 |
| 0.9 | 0.1 | 0.9750 |

Contoh hasil jadwal dari salah satu percobaan diatas dengan jumlah individu sebanyak 400, 7 generasi, nilai probabilitas *crossover* sebesar 0.7 dan probabilitas mutasi sebesar 0.1 adalah sebagai berikut:

Tabel 8 Salah satu contoh jadwal fuzzy relation-algoritma genetika

| Mahasiswa | Penguji 1 | Penguji 2 | Jadwal | Pembimbing 1 |
|-------------------|-----------|-----------|---------------------|--------------|
| Rizki Alfiansyah | KMM | MHD | Sel 08:30 - 10:10 | FTY |
| Melyana Dwitasari | ERW | VIR | Jumat 12:30 - 14:10 | SSI |
| Arin Wahyuningsih | SSI | ADW | Kamis 08:30 - 10:10 | FTY |
| Cici Olivia | ADW | ANQ | Sel 08:30 - 10:10 | IND |
| ... | ... | ... | ... | ... |
| Faishal Mandala P | MHD | TSA | Jumat 08:30 - 10:10 | ADW |

| Pembimbing 2 | Ruang | Fitness | Penalti | Relasi |
|--------------|-------|-------------|---------|--------|
| IHS | 2 | 0.471428571 | 6 | 0.8 |
| ANQ | 2 | 0.475961538 | 12 | 0.875 |

| | | | | |
|-----|-----|-------------|-----|-------------|
| MHD | 3 | 0.767857143 | 0 | 0.535714286 |
| YLS | 1 | 0.847222222 | 0 | 0.694444444 |
| ... | ... | ... | ... | ... |
| ANQ | 2 | 1 | 0 | 1 |

Skenario 3

Pada skenario ketiga ini dilakukan implementasi kasus menggunakan algoritma *fuzzy relation*-algoritma genetika adaptif. Untuk nilai pc dan pm-nya berubah-ubah untuk setiap generasi. Berikut contoh perubahan nilainya:

Tabel 9 Perubahan Pc Pm adaptif

| Generasi ke - <i>i</i> | pc | F_Aksen | pm | F | F_Average | F_Maksmimum |
|------------------------|--------|---------|--------|--------|-----------|-------------|
| 1. | 0.0875 | 0.9712 | 0.5 | 0.4038 | 0.6705 | 1 |
| 2. | 0.1314 | 0.9172 | 0.0876 | 0.9615 | 0.7805 | 1 |
| 3. | 0.1653 | 0.9615 | 0.0827 | 0.9615 | 0.7673 | 1 |
| 4. | 0.1873 | 0.9712 | 0.1249 | 0.9615 | 0.8460 | 1 |
| 5. | 0 | 1 | 0.5 | 0.8654 | 0.8717 | 1 |
| 6. | 0 | 1 | 0.5 | 0.5330 | 0.8573 | 1 |
| 7. | 0 | 1 | 0.1537 | 0.9615 | 0.8749 | 1 |

Pada algoritma genetika adaptif masing-masing parameter dilakukan percobaan sebanyak 10 kali dan diambil nilai rata-ratanya. Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 200, jumlah generasi 14, dapat dilihat pada tabel dibawah:

Tabel 10 Hasil rata-rata fitness fuzzy relation-algoritma genetika adaptif individu 200 generasi 14

| Probabilitas <i>Crossover</i> | Probabilitas Mutasi | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-------------------------------|---------------------|---|
| Adaptif | Adaptif | 0.9760 |

Untuk hasil nilai rata-rata *fitness* maksimum dari masing-masing populasi mahasiswa, dengan jumlah individu sebanyak 400 dan jumlah generasi sebanyak 7, dapat dilihat pada tabel dibawah:

Tabel 11 Hasil rata-rata fitness fuzzy relation-algoritma genetika adaptif individu 400 generasi 7

| Probabilitas <i>Crossover</i> | Probabilitas Mutasi | Rata - Rata Fitness Maksimum Populasi Mahasiswa |
|-------------------------------|---------------------|---|
| Adaptif | Adaptif | 0.9775 |

Contoh hasil jadwal sidang dari salah satu percobaan diatas dengan jumlah individu sebanyak 400 dan 7 generasi adalah sebagai berikut:

Tabel 12 Salah satu contoh jadwal fuzzy relation-algoritma genetika adaptif

| Mahasiswa | Penguji 1 | Penguji 2 | Jadwal | Pembimbing 1 |
|------------------|-----------|-----------|---------------------|--------------|
| I Komang Gede R | JDN | SSI | Kamis 08:30 - 10:10 | RFU |
| Septian Nugraha | VIR | ZKA | Kamis 12:30 - 14:10 | YLS |
| Husna Aydadenta | IZA | KMM | Senin 08:30 - 10:10 | JDN |
| Rizki Alfiansyah | IZA | IMN | Rabu 12:30 - 14:10 | VIR |
| | | | | |
| Irvan Nur Azis | YLS | PHN | Kamis 08:30 - 10:10 | FTR |

| Pembimbing 2 | Ruang | Fitness | Penalti | Relasi |
|--------------|-------|-------------|---------|-------------|
| ANQ | 1 | 0.571428571 | 6 | 1 |
| ERW | 2 | 0.907894737 | 0 | 0.815789474 |
| DTO | 3 | 0.909090909 | 0 | 0.818181818 |
| HIS | 2 | 0.91 | 0 | 0.82 |
| | | | | |
| IMN | 1 | 0.885869565 | 0 | 0.77173913 |

Pada skenario pertama, untuk jumlah individu sebanyak 200 terlihat nilai rata-rata *fitness* maksimum terkecil terdapat pada nilai probabilitas *crossover* sebesar 0.7, dan probabilitas mutasi sebesar 0.06, dengan nilai *fitness* 0.8759. Sedangkan nilai rata-rata *fitness* maksimum terbesar terdapat pada nilai probabilitas *crossover* sebesar 0.5 dan nilai probabilitas mutasi sebesar 0.1, dengan nilai *fitness* 0.8963.

Untuk jumlah individu sebanyak 400, nilai rata-rata *fitness* maksimum terkecil terdapat pada nilai probabilitas *crossover* sebesar 0.6 dan probabilitas mutasi sebesar 0.06, dengan nilai *fitness* 0.9129. Sedangkan nilai rata-rata *fitness* maksimum terbesar terdapat pada nilai probabilitas *crossover* sebesar 0.9 dan nilai probabilitas mutasi sebesar 0.1, dengan nilai *fitness* sebesar 0.9333. Untuk rata-rata *fitness* dari jadwal pada skenario 1 yaitu sebesar 0.8243.

Pada skenario kedua, dilihat hasil akhir dari algoritma *fuzzy relation* untuk mahasiswa yang bernama Choiryaldi Setya Pratama cocok diuji oleh dosen Bapak Adiwijaya dan Bapak Yuliant, namun untuk nilai kecocokannya lebih besar yang Bapak Yuliant daripada Bapak Adiwijaya Hal ini dikarenakan TA dari Choiryaldi Setya Pratama masuk dalam klasifikasi bidang pemrograman terstruktur dan bidang optimasi. Sedangkan Bapak Adiwijaya memiliki nilai relasi terhadap pemrograman terstruktur dan optimasi sebesar 0.8 dan 0.3, dan Bapak Yuliant memiliki nilai relasi sebesar 0.8 dan 0.55. Hal ini juga berlaku untuk mahasiswa yang lain.

Untuk jumlah individu sebanyak 200 nilai rata-rata *fitness* maksimum terkecil terdapat pada nilai probabilitas *crossover* sebesar 0.8, dan probabilitas mutasi sebesar 0.06, dengan nilai *fitness* 0.9617. Sedangkan nilai rata-rata *fitness* maksimum terbesar terdapat pada nilai probabilitas *crossover* sebesar 0.7 dan nilai probabilitas mutasi sebesar 0.1, dengan nilai *fitness* 0.9773.

Untuk jumlah individu sebanyak 400, nilai rata-rata *fitness* maksimum terkecil terdapat pada nilai probabilitas *crossover* sebesar 0.6 dan probabilitas mutasi sebesar 0.06, dengan nilai *fitness* 0.9129. Sedangkan nilai rata-rata *fitness* maksimum terbesar terdapat pada nilai probabilitas *crossover* sebesar 0.6 dan nilai probabilitas mutasi sebesar 0.1, dengan nilai *fitness* sebesar 0.9784. Untuk rata-rata *fitness* dari jadwal pada skenario 2 yaitu sebesar 0.8579.

Pada skenario 3, perubahan nilai p_c bergantung pada nilai p_m dan p_c sesuai pada persamaan (2.12). Jika nilai p_m lebih besar dari nilai p_c , maka syarat pertama yang dijalankan. Jika nilai p_m kurang dari sama dengan p_c , maka syarat kedua yang dijalankan. Hal ini terlihat pada hasil percobaan pada Tabel 4.10, yang dimana setiap generasi nilai p_m lebih besar dari nilai p_c , maka syarat pertama yang dijalankan. Nilai p_c pada algoritma adaptif

berkisar 0 – 1, untuk nilai p_m -nya sebesar 0 – 0.5. Perlu diingat bahwa nilai p_c dan p_m pada kasus ini ditentukan sebesar 1, sedangkan nilai p_c dan p_m ditentukan sebesar 0.5 [12].

Begitupun dengan nilai p_m bergantung pada nilai p_c dan p_m sesuai dengan persamaan (2,13), untuk ketentuannya sama dengan nilai p_c , namun diganti dengan nilai p_m . Pada tabel 4.10 ditunjukkan nilai p_m dari salah satu individu pada populasi mahasiswa. Pada generasi pertama nilai $p_m = 0.9615$, hal ini dikarenakan nilai p_c kurang dari nilai p_m sehingga syarat kedua dari persamaan (2.13) yang dijalankan. Pada generasi kedua nilai p_c lebih besar dari nilai p_m , sehingga syarat pertama yang dijalankan dan hasil perhitungan nilai p_m -nya yaitu sebesar 0.9615, begitu juga untuk seterusnya.

Untuk nilai rata-rata *fitness* maksimum terbesar didapat pada jumlah individu sebanyak 400 dengan nilai *fitness* sebesar 0.9782. Hal ini dikarenakan pada algoritma genetika adaptif, semakin banyak jumlah individu, maka sebaran solusinya semakin luas sehingga peluang untuk mencapai nilai *fitness* maksimum semakin besar. Untuk rata-rata *fitness* dari jadwal pada skenario 3 yaitu sebesar 0.9687.

V. KESIMPULAN

Setelah dilakukan beberapa skenario didapat kesimpulan yaitu Algoritma genetika adaptif dan *fuzzy relation* dapat diterapkan pada kasus penjadwalan sidang Tugas Akhir Fakultas Informatika program studi S1 Ilmu Komputasi, dengan proses implementasinya pertama tentukan relasi masing-masing dosen terhadap mahasiswa menggunakan metode *fuzzy relation*. Relasi ini digunakan untuk menentukan kandidat dosen sebagai penguji terhadap mahasiswa. Lalu tentukan jadwal sidangnya menggunakan metode algoritma genetika adaptif, dengan nilai p_c dan p_m yang berubah-ubah.

Hasil jadwal dari metode algoritma genetika adaptif yang digabung dengan *fuzzy relation* pada percobaan ini memiliki *fitness* yang lebih besar, dengan kata lain jumlah bentrokan lebih sedikit dan dosen penguji sesuai dengan bidang dari TA mahasiswa. Hal ini dikarenakan kandidat penguji dievaluasi dan dipilih dengan melihat seberapa cocok dosen tersebut menguji TA dari seorang mahasiswa.

Pada algoritma genetika adaptif nilai p_c dan p_m berubah sesuai dengan kebutuhan dari individu-individu pada populasi. Jika nilai p_c (nilai *fitness* terbesar pada populasi calon *crossover*) atau p_m (nilai *fitness* dari individu calon mutasi) lebih besar dari (rata-rata *fitness* pada populasi) maka nilai p_c atau p_m nya cenderung mengecil.

Nilai *fitness* terbaik tidak selalu diperoleh dengan nilai probabilitas *crossover* dan mutasi yang besar, hal ini terlihat pada hasil *fitness* algoritma genetika biasa yang digabung dengan algoritma *fuzzy relation* maupun tidak. Untuk nilai *fitness* terbesar didapatkan pada nilai p_c sebesar 0.6, nilai p_m sebesar 0.1, dan jumlah individu sebanyak 400 dengan nilai *fitness* sebesar 0.9784.

REFERENSI

- [4] Pratiwi, Anissa Dian. 2010, *Otimasi Penyusunan Jadwal Sidang Tugas Akhir Dengan Metode Fuzzy Relation Dan Harmony Search Algorithm*, Bandung, Tugas Akhir Teknik Informatika Universitas Telkom.
- [5] Intan, Rolly., Mukaidono, Masao. 2004. *Fuzzy Conditional Probability Relations and their Applications in Fuzzy Information*. Japan: Department of Computer Science, Meiji University.
- [2] Gen, Mitsuo and Runwei., Cheng. 1997. *Genetic Algorithms And Engineering Design*, New York, John Wiley & Sons, Inc.
- [3] Indrianingsih, Yuliani. 2010, *Algoritma Genetik Untuk Menyelesaikan Masalah Optimasi Fungsi Berkendala dengan Pengkodean Bilangan Bulat*, Yogyakarta, Sekolah Tinggi Teknologi Adisutjipto.

- [7] Jha, Sujit Kumar. 2014, *Exam Timetabling Problem Using Genetic Algorithm*, Sultanate of Oman , Engineering Department, Ibra College of Technology.
- [9] Suyanto. 2014. *Artificial Intelligence : Searching, Reasoning, Planning, Learning, Revisi Ke Dua*, Bandung, Informatika.
- [10]Suyanto. 2008, *Soft Computing, Membangun Mesin Ber-IQ Tinggi*, Bandung, Informatika.
- [12]Dey, Dipanjan Kumar. 2014. *Mathematica Study OF Adaptive Genetic Algorithm (AGA) with Mutation and Crossover probabilities*. Departement of Computer Science & Engineering, PITM, Kolkata.