

# Optimasi Penjadwalan Ujian Tugas Akhir Dengan Menggunakan Algoritma Genetika

(Final Project Scheduling Optimization Using Genetic Algorithm)

Adi Panca Saputra Iskandar

Teknik Informatika, STMIK STIKOM Indonesia  
Jl. Tukad Pakerisan No. 97, Denpasar Bali, Indonesia  
Email: adipancaiskandar@gmail.com

**Abstract** - Final Project is one of the requirements for STMIK STIKOM Indonesia students to complete their studies. The final project has two stages, namely the seminar proposal process and the final project session, for this stage of course the study party must schedule a schedule for these stages. The problem that often occurs in scheduling activities is the incidence of clashes between schedules and other schedules, clashes with teaching activities of lecturers as supervisors and examiners. and there is a request for lecturer prohibition time to test. One method to solve this problem is by using an algorithm that works through natural selection and genetics. There are 8 genetic algorithm procedures, coding technique procedures, initial population and chromosome random (random), fitness function to track the number of clashes between schedules, roulette-wheel selection method, crossing over, genetic mutation, elitism, and the condition is complete when the maximum iteration has been reached. The output of the system is in the form of lecture scheduling arrangements and final semester examinations in PDF file format.

**Key words:** Sistem Penjadwalan, Algoritma Genetika, Artificial Intelligence.

## I. PENDAHULUAN

Tugas Akhir merupakan salah satu persyaratan bagi setiap mahasiswa STMIK STIKOM Indonesia untuk menyelesaikan studinya. Salah satu tahapan penting dalam menyelesaikan Tugas Akhir adalah proses seminar proposal dan sidang tugas akhir.

STMIK STIKOM Indonesia merupakan salah satu perguruan tinggi yang berlokasi di Denpasar Bali dan beroperasi menerima mahasiswa baru sejak tahun 2008 [1] telah menerapkan sebuah sistem untuk menunjang hal tersebut. Sistem ini dijalankan dengan melakukan pengarsipan berkas-berkas pengajuan tugas akhir, penentuan dosen pembimbing dan penguji, serta penyusunan jadwal seminar dan sidang tugas akhir. Namun sistem ini masih dijalankan secara manual atau tidak terkomputerisasi. Dalam pelaksanaannya sistem ini melibatkan kepala, sekretaris, dan staf program studi.

Proses penjadwalan adalah suatu proses untuk menerapkan komponen - komponen yang diperlukan seperti ruang, waktu, dosen, dll. Jika menggunakan sistem manual maka masalah ini membutuhkan waktu proses yang cukup lama untuk pencarian solusinya atau

optimasinya, terlebih lagi bila ukuran permasalahan semakin besar dengan bertambahnya jumlah komponen dan tetapan atau syarat yang ditentukan oleh institusi tempat jadwal tersebut di gunakan.

Sistem penjadwalan manual seperti ini memiliki kelemahan khususnya dari segi efisiensi penyusunan jadwal seminar dan sidang tugas akhir memerlukan waktu relatif lama, terlebih lagi ukuran permasalahan yang semakin besar karena bertambahnya jumlah peserta ujian Tugas Akhir pada STMIK STIKOM Indonesia. Tabel I menunjukkan peningkatan rata-rata peserta Tugas Akhir setiap tahunnya berdasarkan hasil observasi.

TABEL I. PENINGKATAN RATA – RATA PESERTA TUGAS AKHIR

| No. | Tahun | Jumlah Peserta | Waktu Penjadwalan |
|-----|-------|----------------|-------------------|
| 1   | 2012  | 35             | 1 Hari            |
| 2   | 2013  | 129            | 3 Hari            |
| 3   | 2014  | 235            | 6 Hari            |
| 4   | 2015  | 328            | 10 Hari           |
| 5   | 2016  | 472            | 14 Hari           |
| 6   | 2017  | 462            | 14 Hari           |
| 7   | 2018  | 481            | 15 Hari           |
| 8   | 2019  | 489            | 15 Hari           |

Tabel I menunjukkan bahwa kendala utama yang dihadapi pada sistem ini terletak pada meningkatnya jumlah peserta yang mempengaruhi waktu penjadwalan sehingga menjadi sebuah masalah yang akan diangkat menjadi bahan penelitian kali ini.

Ada beberapa metode dan algoritma yang sering digunakan dalam menyelesaikan masalah penjadwalan baik perkuliahan maupun ujian yang masing-masing memiliki keunggulan. Salah satu metode dan algoritma tersebut adalah Algoritma Genetika, Algoritma Genetika telah banyak dipelajari dan telah berhasil untuk diimplementasikan pada permasalahan penjadwalan perawat [2], dengan demikian Algoritma Genetika ini akan diterapkan sebagai metode dasar penjadwalan ujian Tugas Akhir pada STMIK STIKOM Indonesia.

Algoritma genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi [2].

Diharapkan dengan digunakannya algoritma genetik akan diperoleh optimasi penjadwalan yang menghasilkan generasi terbaik dari induknya serta menghasilkan sebuah system penjadwalan ujian tugas akhir yang optimal dengan memperhatikan berbagai aspek yang menjadi pertimbangan dan memiliki waktu proses yang lebih cepat dibanding manual.

Berdasarkan uraian yang telah disampaikan maka akan dilakukan penelitian dengan judul Optimasi Penjadwalan Ujian Tugas Akhir dengan Menggunakan Algoritma Genetika pada STMIK STIKOM Indonesia.

## II. TINJAUAN PUSTAKA

Algoritma genetika merupakan evaluasi atau perkembangan dunia komputer dalam bidang kecerdasan buatan (*artificial intelligence*). Kemunculan algoritma genetika ini terinspirasi oleh teori Darwin dan teori-teori dalam ilmu biologi, sehingga banyak istilah dan konsep biologi yang digunakan dalam algoritma genetika, karena sesuai dengan namanya, proses-proses yang terjadi dalam algoritma genetika sama dengan apa yang terjadi pada evaluasi biologi [3].

Algoritma genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional.

Sejak pertama kali dirintis oleh John Holland pada tahun 1960-an, algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma ini banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal [3].

Terdapat beberapa definisi penting dalam Algoritma Genetika yang perlu diperhatikan, yaitu:

- Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa biner, float, integer maupun karakter, atau kombinatorial
- Allele*, merupakan nilai dari gen
- Individu atau kromosom, gabungan gen-gen yang membentuk nilai tertentu dan merupakan salah satu solusi yang mungkin dari permasalahan.
- Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evaluasi.
- Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma genetika.

Adapun Komponen-Komponen dalam Algoritma Genetika yaitu:

### 1. Skema Pengkodean

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetik, maka alternatif solusi tersebut harus dikodekan terlebih dahulu ke dalam

bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengodekan informasi yang disimpan didalam individu atau kromosom. Gen dapat direpresentasikan dalam bentuk: bit, bilangan real, string, daftar aturan, gabungan dari beberapa kode, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

### 2. Membangkitkan Populasi Awal dan Kromosom

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu atau kromosom secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. Teknik dalam pembangkitan populasi awal pada penelitian ini menggunakan metode random search, pencarian solusi dimulai dari suatu titik uji tertentu secara acak. Titik uji tersebut dianggap sebagai alternatif solusi yang disebut sebagai populasi.

### 3. Nilai *Fitness*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya [3]. Didalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, solusi yang akan dicari adalah memaksimalkan fungsi  $h$  (dikenal sebagai masalah maksimasi) sehingga nilai *fitness* yang digunakan adalah nilai dari fungsi  $h$  tersebut, yakni  $f = h$  (di mana  $f$  adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi  $h$  (masalah minimasi), maka fungsi  $h$  tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena itu nilai *fitness* yang bisa digunakan adalah  $f = 1/h$ , yang artinya semakin kecil nilai  $h$ , semakin besar nilai  $f$ . Tetapi hal ini akan menjadi masalah jika  $h$  bisa bernilai 0, yang mengakibatkan  $f$  bisa bernilai tak hingga. Untuk mengatasinya,  $h$  perlu ditambah sebuah bilangan yang dianggap kecil [0-1] sehingga nilai *fitness*-nya menjadi:

$$f = \frac{1}{1+(h)} = 0.5 \quad (1)$$

dengan  $a$  adalah bilangan yang kecil dan bervariasi [0-1] sesuai dengan masalah yang akan diselesaikan.

### 4. Seleksi

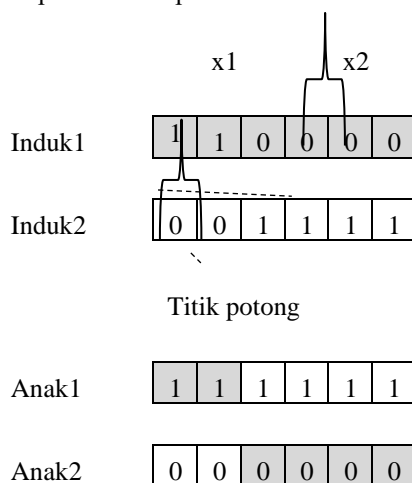
Pembentukan susunan kromosom pada suatu populasi baru biasanya dilakukan secara proporsional sesuai dengan nilai *fitness*-nya [4]. Suatu metode seleksi yang umumnya digunakan adalah *roulette-wheel*. Metode seleksi dengan mesin *roulette* ini merupakan metode yang paling sederhana dan sering dikenal dengan nama *stochastic sampling with replacement*. Cara kerja metode ini adalah sebagai berikut:

- Hitung total *fitness* semua individu

- b) Hitung probabilitas seleksi masing-masing individu
- c) Dari probabilitas tersebut, dihitung jatah interval masing-masing individu pada angka 0 sampai 1
- d) Bangkitkan bilangan random antara 0 sampai 1
- e) Dari bilangan random yang dihasilkan, tentukan urutan untuk populasi baru hasil proses seleksi.

#### 5. Pindah Silang (Crossover)

Salah satu komponen yang paling penting dalam algoritma genetika adalah pindah silang atau crossover [4]. Sebuah kromosom yang mengarah pada solusi yang baik dapat diperoleh dari proses memindah-silangkan dua buah kromosom. Pindah silang juga dapat berakibat buruk jika ukuran populasinya sangat kecil. Dalam suatu populasi yang sangat kecil, suatu kromosom dengan gen-gen yang mengarah pada solusi terbaik akan sangat cepat menyebar ke kromosom-kromosom lainnya [5]. Untuk mengatasi masalah ini digunakan suatu aturan bahwa pindah silang hanya bisa dilakukan dengan suatu probabilitas tertentu, artinya pindah silang bisa dilakukan hanya jika suatu bilangan random yang dibangkitkan kurang dari probabilitas yang ditentukan tersebut. Pada umumnya probabilitas tersebut diset mendekati 1. Pindah silang yang paling sederhana adalah pindah silang satu titik potong (*one-point crossover*). Suatu titik potong dipilih secara acak (*random*), kemudian bagian pertama dari orangtua 1 digabungkan dengan bagian kedua dari orangtua 2 seperti terlihat pada Gambar 1.



Gambar 1. Pindah Silang pada Algoritma Genetika

#### 6. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom [5]. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Metode mutasi yang digunakan adalah mutasi dalam pengkodean nilai. Proses mutasi dalam pengkodean nilai dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan memilih sembarang posisi gen pada kromosom, nilai yang ada tersebut kemudian dirubah dengan suatu nilai tertentu yang diambil secara acak [6].

Contoh:

**Kromosom sebelum mutasi :** 1 3 4 7 6

**Kromosom sesudah mutasi :** 1 2 4 8 6

#### 7. Elitisme

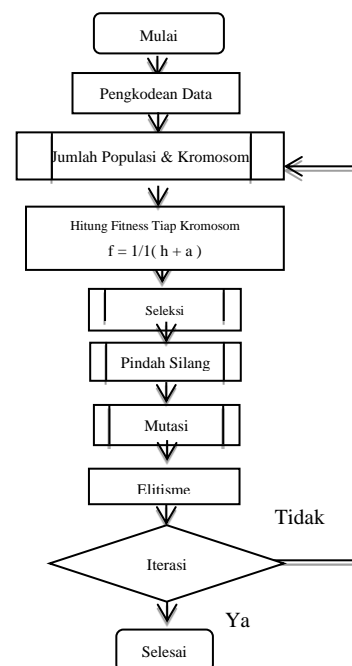
Seleksi dilakukan secara acak (*random*), maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Kalaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*-nya menurun) karena proses pindah silang. Untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa salinannya. Prosedur ini dikenal sebagai Elitisme.

#### 8. Kondisi Selesai

Kondisi selesai yang dapat menghentikan proses algoritma genetika ini adalah jika jumlah generasi atau iterasi maksimum telah tercapai.

### III. METODE PENELITIAN

Penerapan Algoritma Genetika pada penjadwalan ujian tugas akhir dapat diilustrasikan seperti pada Gambar 2.



Gambar 2. Flowchart Penjadwalan Algoritma Genetika

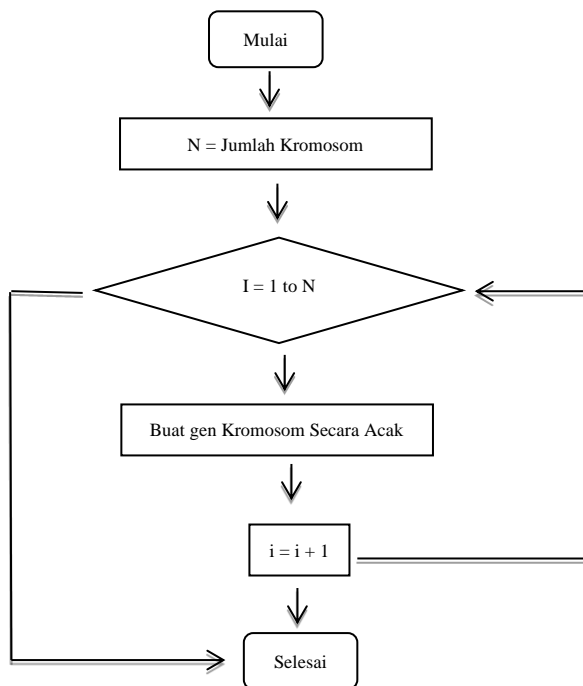
#### 1. Teknik Pengkodean

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom. Masing-masing kromosom berisi sejumlah gen yang mengkodekan informasi yang disimpan didalam kromosom. Pada penelitian ini menggunakan teknik pengkodean dalam bentuk string bit/ varchar yang dipergunakan dalam pemrograman genetika.

#### 2. Populasi dan Kromosom

Menentukan populasi awal adalah proses membangkitkan sejumlah kromosom secara acak

(random). Kromosom menyatakan salah satu alternatif solusi yang mungkin. Kromosom dapat dikatakan sama dengan individu. Ukuran populasi tergantung pada masalah yang akan diselesaikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal dengan cara melakukan inisialisasi solusi yang mungkin kedalam sejumlah kromosom. Panjang satu kromosom ditentukan berdasarkan permasalahan yang diteliti. *Flowchart* nya pada Gambar 3.



Gambar 3. Flowchart Pembentukan Kromosom

Pada penelitian tentang penjadwalan ini solusi yang akan dihasilkan adalah menentukan waktu dan ruang untuk perkuliahan. Panjang satu kromosom adalah gabungan gen berdasarkan jumlah dari seluruh mata kuliah dan kelas yang ditawarkan pada semester aktif. Satu gen berisi informasi waktu dan ruang untuk satu pengajuan tugas akhir dan kelas. Sebagai contoh untuk inisialisasi pembentukan kromosom, misalkan ada sebaran pengajuan Tugas Akhir pada Tabel II, sebaran waktu pada tabel III dan sebaran ruang yang tersedia pada Tabel IV.

Terdapat permintaan dosen githa tidak bisa menguji pada hari senin jam 09.00 dan Manik tidak bisa menguji pada hari selasa jam 09.00. Diasumsikan dalam satu populasi yang terbentuk berjumlah 4 kromosom sesuai dengan jumlah mata kuliah dan kelas yang ada, serta masing-masing kromosom memiliki 4 gen.

Urutan kode pada setiap gen mewakili kode Pengajuan Tugas Akhir, kode waktu, dan kode ruang. Penempatan urutan kode pada setiap gen dilakukan secara acak (*random*) berdasarkan suatu bilangan yang dibangkitkan secara acak (*random*) pula. Pada contoh bilangan tersebut merupakan jumlah dari seluruh matakuliah dan kelas yang ditawarkan.

TABEL II. CONTOH SEBARAN PENGAJUAN TUGAS AKHIR

| No | IdTA | Mahasiswa | Pembimbing | Penguji1 | Penguji2 |
|----|------|-----------|------------|----------|----------|
| 1  | T01  | Adi       | Buda       | Githa    | Aniek    |
| 2  | T02  | Panca     | Dewa Adi   | Manik    | Tyas     |
| 3  | T03  | Saputra   | Gandika    | Arsa     | Ariana   |
| 4  | T04  | Iskandar  | Nyoman S   | Anom     | Jaya     |

TABEL III. CONTOH SEBARAN WAKTU

| No | IdWaktu | Hari   | Waktu         |
|----|---------|--------|---------------|
| 1  | W01     | Senin  | 09:00 – 10:30 |
| 2  | W02     | Senin  | 10:30 – 12:00 |
| 3  | W03     | Selasa | 09:00 – 10:30 |
| 4  | W04     | Selasa | 10:30 – 12:00 |

TABEL IV. CONTOH SEBARAN RUANG

| No | IdRuang | Ruangan   |
|----|---------|-----------|
| 1  | R01     | Ruang 101 |
| 2  | R02     | Ruang 102 |
| 3  | R03     | Ruang 103 |
| 4  | R04     | Ruang 104 |

TABEL V. CONTOH KROMOSOM

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| T02 W03 R01 | T03 W01 R04 | T01 W02 R02 | T04 W04 R03 |
| T02 W03 R02 | T03 W04 R03 | T04 W01 R04 | T01 W02 R01 |
| T01 W01 R02 | T02 W04 R04 | T03 W02 R01 | T04 W03 R03 |
| T04 W04 R03 | T02 W03 R01 | T01 W01 R02 | T03 W02 R04 |

### 3. Fungsi Fitness

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Didalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan *fitness function*. Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Nilai yang dihasilkan oleh fungsi *fitness* merepresentasikan seberapa banyak jumlah persyaratan yang dilanggar, sehingga dalam kasus penjadwalan perkuliahan semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik. Untuk setiap pelanggaran yang terjadi akan diberikan nilai 1. Agar tidak terjadi nilai *fitness* yang tak terhingga maka jumlah total semua pelanggaran akan ditambahkan 1.

$$Fitness \text{ Kromosom } 1 = \frac{1}{1 + (\sum BD) + (\sum BU)} = 1$$

Keterangan :

BD = Banyaknya bentrok dosen mengajar & ujian tugas akhir

BU = Banyaknya bentrok dosen menguji & ujian tugas akhir lainnya.

Dari contoh yang ada akan menghasilkan nilai *fitness* sebagai berikut :

$$\text{Fitness Kromosom 1} = \frac{1}{1 + (1)} = 0.5$$

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

$$\text{Fitness Kromosom 3} = \frac{1}{1 + (2)} = 0.33$$

$$\text{Fitness Kromosom 4} = \frac{1}{1 + (1)} = 0.5$$

#### 4. Seleksi

Pembentukan susunan kromosom pada suatu populasi baru dilakukan dengan menggunakan metode seleksi *roulette-wheel*. Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* dimana masing-masing kromosom menempati potongan lingkaran pada *roulette-wheel* secara proporsional sesuai dengan nilai *fitness*-nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah. Langkah pertama metode ini adalah dengan menghitung total nilai *fitness* seluruh kromosom seperti Tabel VI.

Langkah kedua adalah menghitung probabilitas setiap kromosom dengan cara membagi nilai *fitness* tiap kromosom dengan total nilai *fitness*. Sehingga didapatkan hasil seperti Tabel VII.

Langkah ketiga adalah menempatkan masing-masing kromosom pada interval nilai [0 – 1]. Dapat dilihat pada Tabel VIII.

Untuk menentukan susunan populasi baru hasil seleksi maka dibangkitkan bilangan acak (*random*) antara [0-1]. Dimisalkan bilangan yang dibangkitkan adalah [0,75;0,25;0,9 dan 0,5] maka susunan kromosom populasi baru hasil seleksi adalah sesuai dengan yang terlihat pada Tabel IX.

TABEL VI. NILAI FITNES

| Kromosom                   | Nilai <i>fitness</i> |
|----------------------------|----------------------|
| 1                          | 0,5                  |
| 2                          | 0,5                  |
| 3                          | 0,33                 |
| 4                          | 0,5                  |
| Total Nilai <i>Fitness</i> | 1,83                 |

TABEL VII. PROBABILITAS TIAP KROMOSOM

| Kromosom           | Probabilitas     |
|--------------------|------------------|
| 1                  | 0,5/1,83=0,273   |
| 2                  | 0,5/1,83=0,273   |
| 3                  | 0,33/1,83= 0,181 |
| 4                  | 0,05/1,83= 0,273 |
| Total Probabilitas | 1                |

TABEL VIII. INTERVAL TIAP KROMOSOM

| Kromosom | Interval Nilai |
|----------|----------------|
| 1        | 0-0,273        |
| 2        | 0,273-0,546    |
| 3        | 0,546- 0,727   |
| 4        | 0,727 –1       |

TABEL IX. HASIL SELEKSI

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| T01 W01 R02 | T02 W04 R04 | T03 W02 R01 | T04 W03 R03 |
| T02 W03 R01 | T03 W01 R04 | T01 W02 R02 | T04 W04 R03 |
| T04 W04 R03 | T02 W03 R01 | T01 W01 R02 | T03 W02 R04 |
| T02 W03 R02 | T03 W04 R03 | T04 W01 R04 | T01 W02 R01 |

#### 5. Pindah Silang

Pindah silang (*CrossOver*) digunakan sebagai metode pemotongan kromosom secara acak (*random*) dan merupakan penggabungan bagian pertama dari kromosom induk 1 dengan bagian kedua dari kromosom induk 2. Pindah silang bias dilakukan hanya jika suatu bilangan acak (*random*) yang dibangkitkan untuk kromosom kurang dari probabilitas pindah silang (*Pc*) yang ditentukan. *Pc* umumnya diset mendekati 1, misalnya 0,5. Metode pindah silang yang paling umum digunakan adalah pindah silang satu titik potong (*one - point crossover*) [3]. Suatu titik potong dipilih secara acak (*random*), kemudian bagian pertama dari kromosom induk 1 digabungkan dengan bagian kedua dari kromosom induk 2. Bilangan acak (*random*) yang dibangkitkan untuk menentukan posisi titik potong adalah [1-N] dimana N merupakan banyaknya jumlah gen dalam satu kromosom. Dimisalkan dari contoh yang ada nilai untuk kromosom 2 dan 4 kurang dari *Pc* yang ditetapkan serta bilangan acak (*random*) untuk posisi titik potong adalah pada posisi gen ke-2, maka proses pindah silangnya adalah :

TABEL X. PINDAH SILANG

|                          |                         |             |
|--------------------------|-------------------------|-------------|
| Kromosom 2 = T02 W03 R01 | T03 W01 R04 T01 W02 R02 | T04 W04 R03 |
| Kromosom 4 = T02 W03 R02 | T03 W04 R03 T04 W01 R04 | T01 W02 R01 |

Titik Potong dapat ditunjukkan antara pemisah kolom 2 dan kolom 3 pada Tabel X dan pada Tabel XI.

TABEL XI. HASIL PINDAH SILANG

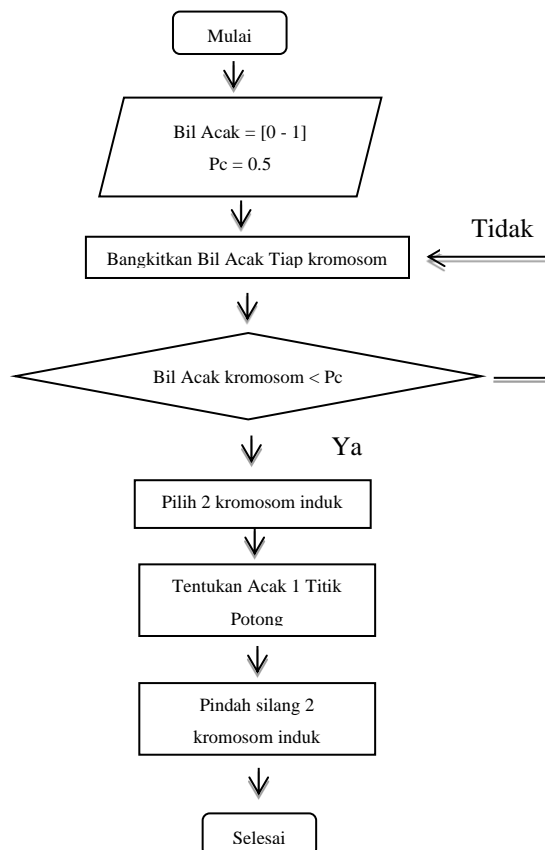
|                                 |                         |             |
|---------------------------------|-------------------------|-------------|
| <b>Kromosom 2</b> = T02 W03 R01 | T03 W04 R03 T04 W01 R04 | T01 W02 R01 |
| <b>Kromosom 4</b> = T02 W03 R02 | T03 W01 R04 T01 W02 R02 | T04 W04 R03 |

Tabel XI merupakan Hasil dari Pindah Silang dari Tabel X:

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

$$\text{Fitness Kromosom 4} = \frac{1}{1 + (1)} = 0.5$$

dari hasil pindah silang pada Tabel XI maka didapatkan nilai *fitness* dari kromosom 2 sebesar 0.5 dan nilai *fitness* dari kromosom 4 0.5 atau proses dari pindah silang ini dapat digambarkan dalam *flowchart* pada Gambar 3 *Flowchart* Pindah Silang.



Gambar 3. Flowchart Pindah Silang

## 6. Mutasi

Proses mutasi adalah suatu proses kemungkinan memodifikasi informasi gen - gen pada suatu kromosom. Perubahan ini dapat membuat solusi duplikasi menjadi memiliki nilai *fitness* yang lebih rendah maupun lebih

tinggi daripada solusi induknya. Jika ternyata diperoleh solusi yang memiliki *fitness* yang lebih tinggi maka hal itulah yang diharapkan. Tetapi jika diperoleh solusi dengan nilai *fitness* yang lebih rendah maka bias jadi pada iterasi berikutnya diperoleh solusi hasil mutasi yang lebih baik nilai *fitness* nya daripada solusi induknya. Untuk semua gen yang ada, jika bilangan acak (random) yang dibangkitkan kurang dari probabilitas mutasi (Pmut) yang telah ditentukan maka beberapa informasi gen akan dirubah dengan menggunakan metode pengkodean nilai. Pmut umumnya diset antara [0– 1], misalnya 0,1 [3].

Untuk mendapatkan posisi gen yang akan dimutasi maka perlu dihitung jumlah total gen dalam satu populasi yaitu :

**Total Gen = Jumlah gen dalam satu kromosom x Jumlah kromosom yang ada.**

Berdasarkan contoh yang ada maka total gen adalah = 4x4 = 16. Probabilitas mutasi ditetapkan 0,1 maka diharapkan mutasi yang terjadi adalah : 0,1 x16 = 1,6 = 2 gen yang akan mengalami mutasi. Selanjutnya dilakukan iterasi sebanyak jumlah total gen [0-16] dan membangkitkan bilangan acak untuk tiap iterasi antara [0-1]. Diasumsikan gen yang mendapatkan bilangan dibawah probabilitas mutasi adalah gen 2 dan 3. Informasi dalam gen yang akan dirubah adalah waktu ujian, maka hasil mutasi pada kromosom tersebut adalah :

**T01 W01 R02    T02 W04 R04    T03 W02 R01    T04 W03 R03**

## Hasil Mutasi

**T01 W04 R02    T02 W02 R04    T03 W03 R01    T04 W01 R03**

Sehingga akan menghasilkan susunan kromosom baru seperti yang ditunjukkan pada Table XII:

TABEL XII. KROMOSOM BARU

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| T01 W04 R02 | T02 W02 R04 | T03 W03 R01 | T04 W01 R03 |
| T02 W03 R01 | T03 W01 R04 | T01 W02 R02 | T04 W04 R03 |
| T04 W04 R03 | T02 W03 R01 | T01 W01 R02 | T03 W02 R04 |
| T02 W03 R02 | T03 W04 R03 | T04 W01 R04 | T01 W02 R01 |

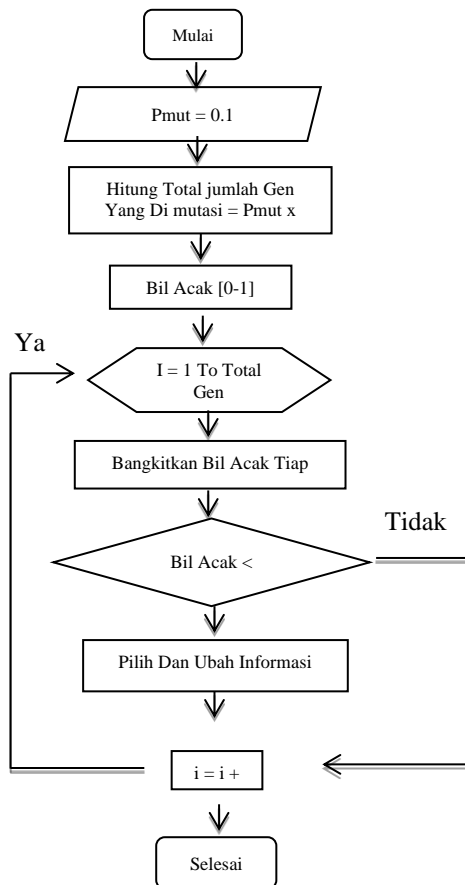
Hasilnya kromosom 1 memiliki nilai *fitness* terbaik karena tidak terdapat pelanggaran yang telah ditetapkan dan merupakan solusi yang diinginkan. *Flowchart* mutasi dapat dilihat pada Gambar 4.

## 7. Elitisme

Proses ini adalah untuk membuat salinan (copy) individu bernilai *fitness* tertinggi agar tidak hilang selama proses evolusi.

## 8. Kondisi Selesai

Kondisi selesai yang dapat menghentikan proses algoritma genetika ini adalah jika jumlah generasi atau iterasi maksimum telah tercapai.



Gambar 4. Flowchart Mutasi

#### IV. IMPLEMENTASI

Berdasarkan hasil pengujian sistem dapat ditulis hasil skenario penjadwalan sebagai berikut :

Langkah awal yang harus dilakukan adalah proses input data calon peserta tugas akhir, hal ini dilakukan oleh mahasiswa yang telah mengambil kartu rencana studi tugas akhir dengan ketentuan mahasiswa mengajukan judul kepada pihak program studi, setelah itu pihak program studi akan meng-input-kan data yang diperlukan oleh sistem, data yang berhasil diinputkan oleh sistem akan tersimpan di *database* kemudian mahasiswa akan melakukan pengajuan ujian proposal tugas akhir dan kemudian program studi akan memilih pengujian tugas akhir dan data akan segera masuk antrian penjadwalan algoritma genetika, kemudian data akan dikombinasikan dari beberapa tabel yang sudah berelasi dan dijadikan satu populasi dengan susunan kromosom seperti yang ditunjukkan pada Table XIII:

TABEL XIII. HASIL INPUT TUGAS AKHIR

| IdTA | NIM      | Pembimbing | Penguji1   | Penguji2   |
|------|----------|------------|------------|------------|
| 1    | 11101420 | 0820068402 | 0819018901 | 0824048801 |
| 2    | 09101203 | 0820068402 | 0831038302 | 0806068602 |
| 3    | 08103030 | 0820068402 | 0828068501 | 0208048301 |
| 4    | 10101265 | 0814088502 | 0806088501 | 0818118602 |

Kromosom pada sistem dapat ditampilkan seperti yang ditunjukkan pada Gambar 5.

| Adi Panca Saputra Iskandar   11101420   TI-MTI   |                                       |
|--|---------------------------------------|
| Optimasi Penjadwalan Ujian Tugas Akhir Dengan Menggunakan Algoritma Genetika Pada STMIK STIKOM Indonesia                         |                                       |
| Topik  | : Artificial Intelligence             |
| Tahun Ajaran   | : 2014/2015                           |
| Semester   | : Ganjil                              |
| Pembimbing 1   | : I Putu Gede Budayasa, M.T.I.        |
| Pembimbing 2   | : I Made Gede Sri Artha, S.T.         |
| Telepon  | : 085737530302                        |
| Penguji 1  | : Ida Bagus Ari Indra Iswara, M.Kom.  |
| Penguji 2  | : Gede Aditra Pradnyana, M.Kom.       |
| Waktu Proposal   | : 01-01-1970 (00:00:00-00:00:00 WITA) |
| <a href="#">Terima</a> <a href="#">Ubah</a> <a href="#">Proposal</a> <a href="#">Surat Tugas</a> <a href="#">Surat Keputusan</a> |                                       |

Gambar 5. Tampilan Bentuk Kromosom Dalam Sistem

Dari populasi pada Tabel XIII terdiri dari 4 kromosom dengan 5 gen dari masing masing kromosom, dari data tersebut nim mewakili atribut mahasiswa lainnya, nidn mewakili atribut dosen yang meliputi kolom pembimbing, penguji1 dan penguji2, dengan demikian kromosom tersebut akan di cari *constraint* dengan cara menyebarkan hari dan waktu ke setiap kromosom secara *random* maka akan menjadi populasi baru dengan susunan kromosom seperti yang ditunjukkan pada Tabel XIV :

TABEL XIV. SEBARAN WAKTU

| IdTA | NIM      | Pembimbing | Penguji1   | Penguji2   | IdWaktu |
|------|----------|------------|------------|------------|---------|
| 1    | 11101420 | 0820068402 | 0819018901 | 0824048801 | T04     |
| 2    | 09101203 | 0820068402 | 0831038302 | 0806068602 | T01     |
| 3    | 08103030 | 0820068402 | 0828068501 | 0208048301 | T03     |
| 4    | 10101265 | 0814088502 | 0806088501 | 0818118602 | T02     |

*Constraint* pada sistem dapat ditampilkan seperti pada Gambar 6.

| IdWaktu | NIM      | Pembimbing | Penguji1   | Penguji2   |
|---------|----------|------------|------------|------------|
| T04     | 11101420 | 0820068402 | 0819018901 | 0824048801 |
| T01     | 09101203 | 0820068402 | 0831038302 | 0806068602 |
| T03     | 08103030 | 0820068402 | 0828068501 | 0208048301 |
| T02     | 10101265 | 0814088502 | 0806088501 | 0818118602 |

Gambar 6. Tampilan Constraint Dalam Sistem

Dari populasi baru pada Tabel XIV muncul kolom baru dengan nama IdWaktu dimana IdWaktu mewakili atribut waktu seperti hari dan jam. Kemudian dari data tersebut akan dicocokkan dengan tabel kelas, tabel kelas merupakan tabel yang menyimpan data *constraint* sebagai berikut :

TABEL XV. CONSTRAINT KELAS

| IdKelas | NIDN       | Hari  | Jam           |
|---------|------------|-------|---------------|
| 1       | 0820068402 | Senin | 09:00 – 10:30 |
| 2       | 0831038302 | Senin | 12:00 – 14:15 |
| 3       | 0828068501 | Rabu  | 11:00 – 12:30 |
| 4       | 0814088502 | Kamis | 14:15 – 15:45 |

Dari Tabel XV maka populasi baru akan di-*cross check* dengan tabel di kelas apabila ada waktu penjadwalan yang terbentur dengan waktu pada kelas maka populasi baru akan di *cross over* atau dipindah silangkan dengan kromosom lainnya dan akan dicek kembali berdasarkan nilai *fitness* dari nilai *fitness* itu maka kromosom akan diseleksi menggunakan *roulette wheels* setelah itu kromosom akan dihitung probabilitas *cross over* dan kromosom yang memiliki probabilitas yang mencukupi akan disusun kembali kromosom baru secara *descending* atau berdasarkan *fitness* tertinggi dan akan kembali di cek nilai *fitness*-nya jika nilai *fitness* belum maksimal atau masih terjadi bentrokan jadwal maka proses akan dilanjutkan ke tahap selanjutnya yaitu mutasi dengan cara menukar setiap gen yang bermasalah dengan populasi lainnya. Setelah demikian maka setiap generasi akan disimpan dan proses selanjutnya adalah elitism dengan menyimpan susunan kromosom hasil genetika dan pada proses akhir adalah sistem akan mencari *fitness* tertinggi atau yang paling kecil kemungkinan terjadinya bentrokan. Table XVI menunjukkan hasil percobaan dari sepuluh generasi terakhir dari dengan komputasi, dalam percobaan tersebut menghabiskan waktu selama 1 Menit 28 Detik dengan jumlah generasi 1000 dan kromosom acak sebanyak 16 dari 4 populasi yang dibangun, dimana setiap generasi menghasilkan nilai *fitness* yang berbeda beda, jika dilihat pada generasi ke 998 seharusnya nilai *fitness* sudah tertinggi namun proses genetika tidak akan berhenti jika generasi yang dilakukan iterasi tidak mencapai generasi N, untuk itu generasi yang terbaik bisa saja muncul pada generasi tengah, depan maupun belakang, selain itu nilai *fitness* yang paling tinggi belum tentu merupakan generasi terbaik.

TABEL XVI. HASIL SAMPLE PERCOBAAN KOMPUTASI

| UG   | K1 | K2 | K3 | K4 | K5 | NF     |
|------|----|----|----|----|----|--------|
| 991  | 2  | 1  | 3  | 1  | 0  | 0.0082 |
| 992  | 1  | 2  | 1  | 3  | 0  | 0.0082 |
| 993  | 1  | 1  | 0  | 0  | 0  | 0.3324 |
| 994  | 3  | 0  | 0  | 0  | 0  | 0.2441 |
| 995  | 1  | 0  | 0  | 1  | 0  | 0.3324 |
| 996  | 1  | 0  | 0  | 0  | 0  | 0.4183 |
| 997  | 1  | 0  | 1  | 2  | 0  | 0.1719 |
| 998  | 0  | 0  | 0  | 0  | 0  | 0.4803 |
| 999  | 2  | 0  | 1  | 0  | 0  | 0.2441 |
| 1000 | 0  | 0  | 0  | 0  | 0  | 0.4909 |

Keterangan :

UG = Urutan Gen

K1 = Jumlah bentrok jadwal uji dosen

K2 = Jumlah bentrok ruangan uji

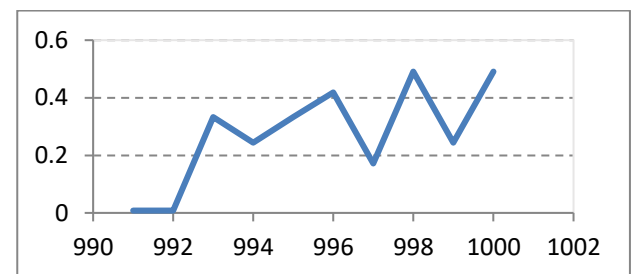
K3 = Jumlah bentrok jadwal uji dengan jadwal mengajar

K4 = Jumlah bentrok jadwal uji dengan constraint dosen

K5 = Jumlah jadwal uji melebihi maksimum dalam sehari

NF = Nilai *Fitness*

Hasil pada Tabel XVI merupakan 10 sample terakhir dari 1000 generasi, pada Tabel XVI diambil 10 sample terakhir dengan tujuan agar tabel yang ditampilkan tidak terlalu banyak karena 10 generasi terakhir ini sudah mewakili variasi generasi yang diproses. Jika direpresentasikan 10 sample terakhir dari 1000 generasi, maka akan menghasilkan grafik yang ditunjukkan pada Gambar 7.



Gambar 7. Grafik Percobaan Nilai Fitness

Dengan demikian jadwal berhasil diproses dengan algoritma genetika. Dalam hal ini proses penjadwalan dilakukan dengan estimasi waktu sekitar kurang lebih 10 menit. Setelah itu mahasiswa akan melanjutkan ke tahap ujian selanjutnya yaitu ujian siding tugas akhir dengan tahapan yang sama seperti ujian proposal. Berdasarkan hasil pengujian pada sistem penjadwalan ini dapat dibandingkan lama waktu penyelesaian penjadwalan secara manual dan genetika sebagai berikut :

TABEL XVII. HASIL PENJADWALAN KESELURUHAN

| No. | Jumlah Peserta | Waktu Penjadwalan Manual | Waktu Penjadwalan Genetika |
|-----|----------------|--------------------------|----------------------------|
| 1   | 35             | 1 Hari                   | 2 Menit                    |
| 2   | 129            | 3 Hari                   | 3 Menit                    |
| 3   | 235            | 6 Hari                   | 6 Menit                    |
| 4   | 328            | 10 Hari                  | 6 Menit                    |
| 5   | 472            | 14 Hari                  | 6 Menit                    |
| 6   | 462            | 14 Hari                  | 6 Menit                    |
| 7   | 481            | 15 Hari                  | 6 Menit                    |
| 8   | 489            | 15 Hari                  | 6 Menit                    |

Hasil output dari penjadwalan dapat ditampilkan seperti yang ditunjukkan pada Gambar 8 :



Jadwal Ujian Proposal Tugas Akhir

| No | NIM      | Nama Mahasiswa                | Pembimbing                           | Penguji I                            | Penguji II                        | Tanggal    | Waktu    | Waktu    | Ruang |
|----|----------|-------------------------------|--------------------------------------|--------------------------------------|-----------------------------------|------------|----------|----------|-------|
| 1  | 12101385 | Ketut Jaya Atmaja             | I Made Andri Pridyana, MT.           | Aniek Suryanti Kusuma, MKom.         | Dwi Putri Otha, MT.               | 12-01-2015 | 13:00:00 | 14:30:00 | 404   |
| 2  | 10101126 | Ida Bagus Putu Purnama Wibawa | I Made Andri Pridyana, MT.           | I Dewa Made Adi Budana Jati, MKom.   | Ayu Masik Dipayana, MMT.          | 12-01-2015 | 14:30:00 | 16:00:00 | 404   |
| 3  | 10101067 | I Made Jansarta               | Aniek Suryanti Kusuma, MKom.         | Brigida Arie Miasartiningtyas, MKom. | Desak Made Dwi Utami Putra, MKom. | 12-01-2015 | 19:30:00 | 21:00:00 | 403   |
| 4  | 11101290 | Titus Agastizetrisa Lala      | I Nyoman Agus Pajanditya, MSn.       | I Dewa Made Adi Budana Jati, MKom.   | I Nyoman Agus Surya Putra, MSn.   | 13-01-2015 | 10:30:00 | 12:00:00 | 401   |
| 5  | 10101284 | I Wayan Agus Paimakara        | Brigida Arie Miasartiningtyas, MKom. | I Made Marbana Yusa, M.Ds.           | Ida Bagus Ari Indra Irawan, MKom. | 13-01-2015 | 14:30:00 | 16:00:00 | 403   |
| 6  | 10101195 | Eddy Soyyun Syah              | I Nyoman Agus Surya Putra, MSn.      | Ida Bagus Ari Indra Irawan, MKom.    | I Nyoman Jayanegara, MSn.         | 14-01-2015 | 16:00:00 | 17:30:00 | 401   |
| 7  | 10101396 | I Nyoman Yoga Sasudana        | I Nyoman Agus Surya Putra, MSn.      | Aniek Suryanti Kusuma, MKom.         | Ode Aditra Pridyana, MKom.        | 14-01-2015 | 17:30:00 | 19:00:00 | 404   |
| 8  | 10101072 | Putu Septia Puri Rahayu       | I Nyoman Agus Surya Putra, MSn.      | I Made Marbana Yusa, M.Ds.           | Ode Aditra Pridyana, MKom.        | 15-01-2015 | 09:00:00 | 10:30:00 | 402   |
| 9  | 10102367 | Eko Febryanto                 | I Made Andri Pridyana, MT.           | I Nyoman Budi Hartawan, MKom.        | I Wayan Sudana, S.T., MKom.       | 15-01-2015 | 10:30:00 | 12:00:00 | 403   |

Gambar 8. Screenshot Hasil Penjadwalan Final dalam PDF

## V. KESIMPULAN DAN SARAN

Proses optimasi penjadwalan seminar dan sidang tugas akhir di STMIK STIKOM Indonesia diawali dengan merancang dan membangun sebuah sistem informasi tugas akhir. Berdasarkan pembahasan hasil yang dicapai maka dapat ditarik kesimpulan sebagai berikut :

1. Optimasi penjadwalan seminar proposal dan sidang tugas akhir pada sistem informasi tugas akhir STMIK STIKOM Indonesia diterapkan dengan melalui tahapan pengkodean, menentukan populasi awal dan inialisasi kromosom, fungsi *fitness*, proses seleksi, pindah silang (cross over), mutasi, hingga elitisme.
2. Semakin banyak generasi atau jumlah peserta ujian tugas akhir yang ditentukan maka semakin lama penjadwalan dilakukan, semakin sedikit generasi atau jumlah peserta yang ditentukan maka semakin cepat jadwal dihasilkan.
3. Hasil penjadwalan akan semakin baik jika generasi yang dibentuk banyak dan sedikit adanya constraint dosen.
4. Sulitnya pengujian manual dan otomatis secara sistem karena adanya fungsi random atau pemberian nilai secara acak yang mengakibatkan hasil perhitungan manual berbeda dengan otomatis secara sistem.
5. Penggunaan algoritma genetika ini merupakan salah satu metode optimasi yang cukup baik namun memiliki kekurangan dengan terjebaknya sebuah generasi pada tingkat maksimum *fitness*, padahal jika generasi yang diambil bukan *fitness* tertinggi mungkin saja bisa lebih optimal.

## DAFTAR PUSTAKA

- [1] Iskandar, Adi Panca dan Supartha, I Kadek Dwi Gandika. 2019. Sistem Informasi Geografis Pemetaan Sebaran Alumni Pada STMIK STIKOM Indonesia. J-COSINE, Vol. 3, No. 2, Desember 2019.
- [2] Putra, Desak Made Dwi Utami dan subanar. 2012. Penerapan Algoritma Genetika Untuk Menyelesaikan Permasalahan Perawat dengan Fuzzy Fitness Function.[ Application of Genetic Algorithm for Solving Problems with Fuzzy Nurses Fitness Function] Yogyakarta: Universitas Gajah Mada.
- [3] Suyanto (2011), Artificial Intelligence (Searching, reasoning, planning, learning) Yogyakarta: Informatika.
- [4] Puspaningrum, Wiga Ayu. Djunaidy, Arif. dan Vinarti , Retno Aulia. 2013. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS [Course Scheduling Using Genetic Algorithm in the Department of Information Systems ITS]. Surabaya: Institut Teknologi Sepuluh November.
- [5] Siswono, Teno dan Palgunadi, Sarngadi. 2014. Analisa Kombinasi Algoritma Genetika dengan Algoritma Palgunadi untuk penjadwalan mata kuliah di Universitas Sebelas Maret [Analysis of Combination of Genetic Algorithm with Palgunadi algorithm for scheduling courses at the University of March]. Surakarta: Universitas Sebelas Maret
- [6] Anamisa , Devie Rosa dan Djunaidy, Arif. 2014. Penyelesaian Penjadwalan Matakuliah Menggunakan Hibridisasi Algoritma Genetika Dan Algoritma Koloni Semut [Completion Course Scheduling Using Genetic Algorithms Hybridization And Ant Colony Algorithm]. Surabaya: Institut Teknologi Sepuluh November