

# PENJADWALAN MATA KULIAH MENGUNAKAN ALGORITMA GENETIKA DI STT WASTUKANCANA PURWAKARTA

Umar Hasan<sup>1</sup>, Teguh Iman Hermanto<sup>2</sup>, M. Rafi Muttaqin<sup>3</sup>

Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukancana Purwakarta

Jalan Cikopak No.53, Mulyamekar, Babakancikao, Purwakarta, Jawa Barat 41151

[umarhasan@stt-wastukancana.ac.id](mailto:umarhasan@stt-wastukancana.ac.id)<sup>1</sup>, [teguh@stt-wastukancana.ac.id](mailto:teguh@stt-wastukancana.ac.id)<sup>2</sup>,

[rafi@stt-wastukancana.ac.id](mailto:rafi@stt-wastukancana.ac.id)<sup>3</sup>

## ABSTRAK

*Generating a study at STT Wastukancana Purwakarta is strongly important to get an effective way to study. As the allocation of rooms to the schedule made is still manual, it is possible there are some room service clashes. To solve that problem, a study scheduling using computerization is needed in order to make no clash of room service. The use of genetic algorithm to solve the problem is able to get the optimal solution in generating the schedule with chromosome representative as an integer from each of data primary key, the beginning of population initialization, the selection with rank-selection method, two-points crossover method, and mutation. From the test, the result points out that the optimal schedule with 416 pengampu data is generated when a number of population are 30, a number of generations are 150, the crossover probability value (Pc) is 0.4, and the mutation probability value (Pm) is 0.37.*

Keywords : *Genetic Algorithm, Crossover Probability, Mutation Probability*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Penjadwalan merupakan salah satu kegiatan penting dalam suatu lembaga. Penjadwalan adalah pengaturan waktu dari satu kegiatan operasi, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi. Dalam suatu lembaga pendidikan, penjadwalan diperlukan antara lain untuk mengalokasikan ruang kelas, peralatan mengajar, tenaga pengajar, staf administrasi, dan pendaftaran penerimaan mahasiswa baru atau ujian [1].

Kegiatan belajar dan mengajar merupakan aktifitas akademik yang paling penting dalam proses transfer ilmu pengetahuan mengenai suatu mata kuliah dengan mengacu jadwal yang tidak terjadi bentrok ruangan dalam penggunaannya. Itu merupakan hal yang diinginkan oleh pihak universitas dalam perkuliahan. Akan tetapi, kenyataan di lapangan tidaklah mudah. Banyaknya aspek yang harus diperhatikan dan dilibatkan, misalnya jam dosen untuk mengajar tidak boleh adanya jadwal mata kuliah yang bentrok pada saat yang bersamaan, tidak boleh adanya bentrok ruang kelas yang dipakai, dan lain-lain.

Sekolah Tinggi Teknologi (STT) Wastukancana Purwakarta merupakan suatu lembaga yang memiliki visi dan misi dalam mencetak mahasiswa yang ideal. STT Wastukancana Purwakarta terletak di daerah Cikopak, Kecamatan Babakancikao, Kabupaten Purwakarta, Jawa Barat. Tahun ajaran 2017-2018 semester genap terdapat 80 dosen yang ditugaskan mengajar di semester tersebut, terdapat 2 ruang untuk praktikum dan 28 ruang untuk teori, terdapat 64 kelas dari 5 jurusan; jurusan teknik informatika, jurusan teknik industri, jurusan teknik mesin, jurusan teknik tekstil, dan manajemen industri, dan terdapat 416 data pengampu dimana data pengampu merupakan data jadwal yang belum memiliki alokasi ruang, jam, dan hari. Kegiatan perkuliahan dimulai jam 08 pagi sampai jam 2 siang atau jam 3 siang untuk kelas pagi, dan perkuliahan berlangsung jam 18.40 sampai jam 22.40 untuk kelas malam.

Berlangsungnya kegiatan pembelajaran di STT Wastukancana ini mengacu terhadap penjadwalan mata kuliah yang dibuat oleh pihak akademik secara manual. Di dalam pembuatan jadwal untuk

perkuliahan sering kali mengalami bentrok jadwal dalam hal alokasi ruangan yang akan digunakan dan jadwal yang belum diberikan alokasi ruang dan waktu dibuat oleh tiap prodi jurusan.

Algoritma genetika adalah pencarian heuristik yang meniru proses evolusi alam. Heuristik ini secara rutin digunakan untuk menghasilkan solusi optimal yang berguna untuk masalah optimasi dan pencarian. Algoritma genetika termasuk kelas yang lebih besar dari algoritma evolusioner, yang menghasilkan solusi untuk masalah optimasi menggunakan teknik yang terinspirasi oleh evolusi alam, seperti warisan, mutasi, seleksi, dan *crossover* [2].

Ada tiga keuntungan utama dalam mengaplikasikan algoritma genetika pada masalah-masalah optimasi [3], yaitu algoritma genetika tidak memerlukan kebutuhan matematis banyak mengenai masalah optimasi, kemudahan dan kenyamanan pada operator-operator evolusi membuat algoritma genetika sangat efektif dalam melakukan pencarian global, dan algoritma genetika menyediakan banyak fleksibilitas untuk digabungkan dengan metode heuristik yang tergantung domain untuk membuat implementasi yang efisien pada masalah-masalah khusus.

Tentu saja, algoritma genetika bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk mencari penyelesaian dari fungsi analitis convex yang “berperilaku baik” yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetika karena metode ini dengan cepat menemukan solusi minimum ketika algoritma genetika masih menganalisa bobot dari populasi awal.

## 1.2 Maksud dan Tujuan

Berdasarkan pemaparan permasalahan penjadwalan kuliah di STT Wastukencana Purwakarta maka penulis mengambil penelitian terhadap pembuatan jadwal kuliah menggunakan algoritma genetika dengan mengacu data perkuliahan semester genap tahun ajaran 2017-2018. Jadwal yang dibuat dengan algoritma genetika diharapkan mampu memberikan solusi terhadap masalah di atas.

## 2. LANDASAN TEORI

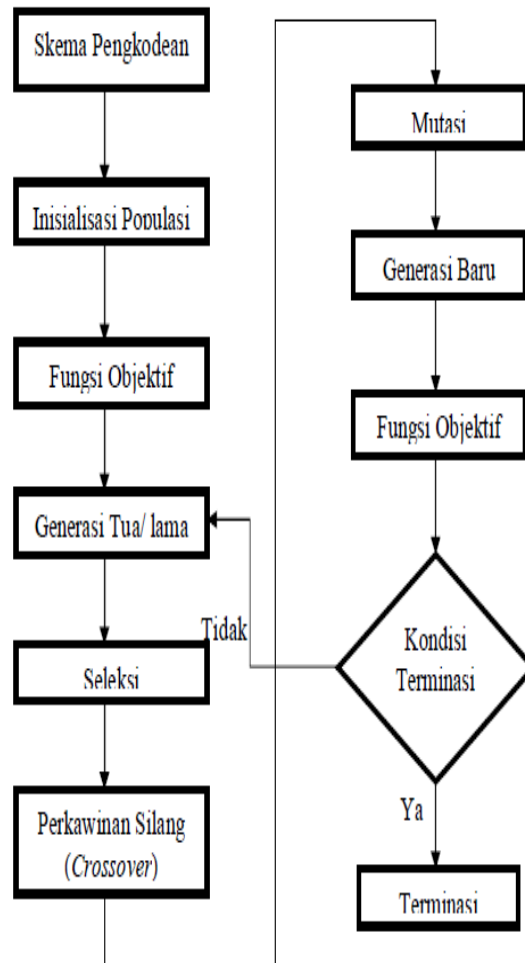
### a. Penjadwalan

Penjadwalan (*scheduling*) merupakan salah satu kegiatan penting dalam perusahaan. Penjadwalan adalah pengaturan waktu dari satu kegiatan operasi, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi. Dalam suatu lembaga pendidikan, penjadwalan diperlukan antara lain untuk mengalokasikan ruang kelas, peralatan mengajar, tenaga pengajar, staf administrasi, dan pendaftaran penerimaan mahasiswa baru atau ujian. Penjadwalan bertujuan meminimalkan waktu proses, waktu tunggu langganan, dan tingkat persediaan, serta penggunaan yang efisien dari fasilitas, tenaga kerja, dan peralatan [1].

### b. Algoritma Genetika

Algoritma genetika ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland pada tahun 1975 melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg pada tahun 1989. Dimana mendefinisikan algoritma genetika ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi dan genetik alam [4].

Algoritma genetika adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner [4]. Struktur umum algoritma genetika dapat dilihat di gambar 1.



Gambar 1 Struktur Umum Algoritma Genetika [5]

Beberapa komponen pada algoritma genetika yaitu [5]:

a. Pengkodean kromosom

Teknik pengkodean adalah bagaimana mengkodekan gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dikodekan terlebih dahulu ke dalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengkodekan informasi yang disimpan di dalam individu atau kromosom. Gen dapat direpresentasikan dalam bentuk bit, bilangan ril, *string*, daftar aturan, gabungan dari beberapa kode, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

b. 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.

c. Nilai *fitness*

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang akan 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  $f=h$  (dimana  $f$  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.

d. Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya [6]. Ada beberapa metode untuk memilih kromosom yang sering digunakan antara lain adalah *roulette wheel selection*, *rank selection*, dan *tournament selection*.

e. Crossover

*Crossover* bertujuan menambah keanekaragaman *string* dalam populasi dengan penyilangan antar-*string* yang diperoleh dari sebelumnya. Beberapa jenis *crossover* adalah *crossover* 1-titik dapat dilihat pada Tabel 1 untuk contohnya, *crossover* 2-titik dapat dilihat pada Tabel 2 untuk contohnya, dan *crossover* seragam dapat dilihat pada Tabel 3 untuk contohnya.

Tabel 1 Crossover 1-titik

|                     |                 |
|---------------------|-----------------|
| Kromosom Orangtua 1 | <b>11001011</b> |
| Kromosom Orangtua 2 | <b>11011111</b> |
| Keturunan           | <b>11001111</b> |

Tabel 2 Crossover 2-titik

|                     |                 |
|---------------------|-----------------|
| Kromosom Orangtua 1 | <b>11001011</b> |
| Kromosom Orangtua 2 | <b>11011111</b> |
| Keturunan           | <b>11011111</b> |

Tabel 3 Crossover seragam

|                     |                 |
|---------------------|-----------------|
| Kromosom Orangtua 1 | <b>11001011</b> |
| Kromosom Orangtua 2 | <b>11011111</b> |
| Keturunan           | <b>11011111</b> |

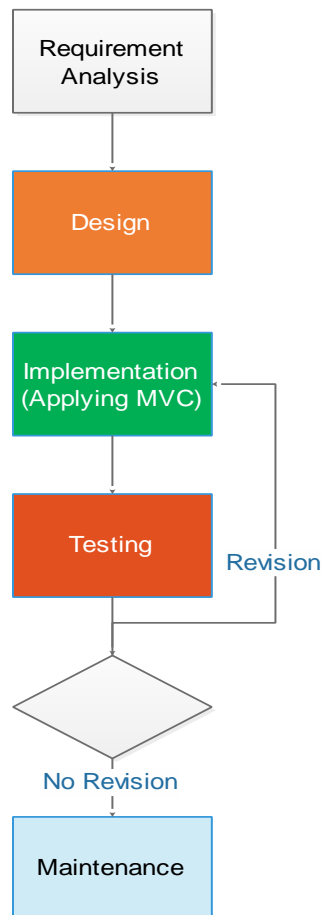
f. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan *fitness* yang tinggi saja, konvergensi *premature*, yaitu mencapai solusi yang optimal lokal sangat mudah terjadi.

### 3. METODOLOGI PENELITIAN

Penelitian ini termasuk dalam bidang rekayasa perangkat lunak. Metode pengembangan sistem yang digunakan adalah *System Development Life Cycle (SDLC) with Waterfall Method* yang dapat digabungkan dengan *Model-View-Controller (MVC) architecture*.

Penulis menggunakan metode pengembangan sistem tersebut karena mengacu terhadap apa yang telah diteliti oleh Hardyanto dkk yang menunjukkan bahwa proses pengembangan dengan metode tersebut menjadi optimal [7]. Gambar 2 menunjukkan gambaran proses dalam *SDLC with Waterfall Method* dengan *MVC architecture*.



Gambar 2 Proses SDLC dengan arsitektur MVC [7]

Beberapa langkah yang dilakukan dalam penelitian ini diantaranya:

1. Studi Pustaka dan *Requirement Analysis*. Di dalam fase ini, menganalisa proses yang sedang berjalan dengan melakukan diskusi dan mengumpulkan dasar-dasar teori dan sumber acuan mengenai algoritma genetika dan data perkuliahan di STT Wastukencana Purwakarta tahun ajaran 2017-2018 semester genap yang nantinya dapat dipelajari mengenai informasi, pengetahuan, dan teori yang terkait tentang pembuatan jadwal perkuliahan dengan algoritma genetika agar dari hasil penelitian dapat menghasilkan penjadwalan kuliah yang optimal.
2. Desain Algoritma dan Desain Tampilan. Di dalam fase ini, merancang algoritma dengan menggunakan algoritma genetika dan merancang pemodelan visual dengan *Unified Modeling Language* (UML). Ada lima tahap di dalam algoritma genetika untuk mendapatkan hasil yang optimal terhadap permasalahan yang ada menurut [2], yaitu pengkodean kromosom, inialisasi populasi, evaluasi fungsi *fitness* dan seleksi, pindah silang, dan mutasi. Perancangan tampilan disajikan dengan empat bentuk diagram pemodelan dengan menggunakan perangkat lunak *Enterprise Architect* dan *Edraw Max*, yakni *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.
3. Implementasi Perancangan dengan arsitektur MVC. Di dalam fase ini, hasil perancangan algoritma dan desain tampilan yang sudah dibuat diimplementasikan ke dalam penulisan kode dengan arsitektur MVC dengan menggunakan perangkat lunak *PHP*, *CodeIgniter Framework*, *Mozilla Firefox*, dan *Xampp*.
4. Pengujian Sistem. Di dalam fase ini, pengujian sistem dilakukan dengan parameter algoritma genetika, yaitu antara lain pengujian ukuran populasi, banyaknya generasi, nilai probabilitas *crossover*, dan nilai probabilitas mutasi. Pengujian dilakukan agar dapat membuktikan bahwa hasil sistem berupa perangkat lunak tersebut telah mampu bekerja dengan baik sesuai kebutuhan dan

mengetahui kinerja algoritma genetika dalam membuat jadwal guna menyelesaikan permasalahan penjadwalan kuliah di STT Wastukencana Purwakarta.

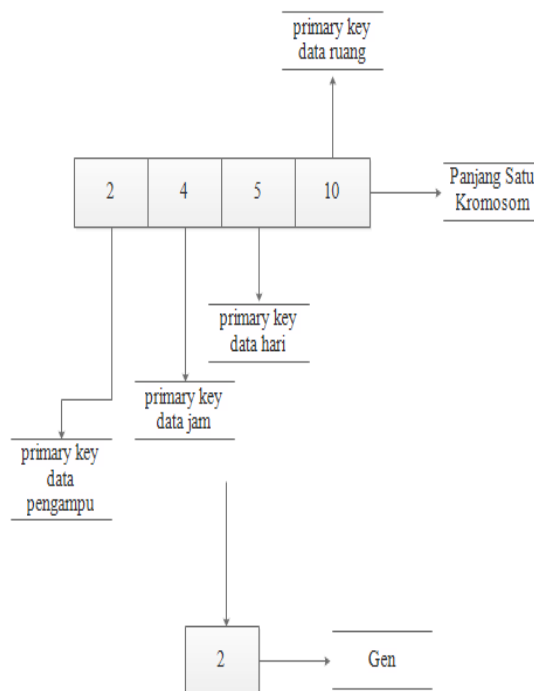
5. Evaluasi Sistem. Di dalam fase ini, mengevaluasi hasil jadwal atau hasil pengujian yang telah dibuat mengenai pembuatan jadwal kuliah menggunakan algoritma genetika. Evaluasi sistem yang dilakukan yaitu melakukan pengujian kembali akan hasil dari parameter terbaik yang telah dilakukan sebelumnya.

#### 4. ANALISIS DESAIN

##### 4.1 Analisis Desain Algoritma

Gambar 4 menunjukkan alur algoritma genetika yang digunakan dalam penjadwalan mata kuliah dengan perulangan yang disesuaikan dengan parameter banyaknya generasi.

1. Pengkodean Kromosom. Di dalam tahap ini, untuk pengkodean kromosom atau representasi kromosom dibutuhkan slot waktu perkuliahan. Pada kasus ini waktu kuliah adalah satu sesi dalam perkuliahan di setiap ruang untuk setiap harinya. Diasumsikan bahwa perkuliahan diselenggarakan pada jam 8 pagi sampai jam 2 siang atau jam 3 siang, kemudian berlanjut di jam 18.40 sampai jam 22.40. Total waktu untuk kelas pagi adalah 6 jam dan total waktu untuk kelas malam adalah 4 jam. Perkuliahan berlangsung mulai hari Senin sampai dengan hari Sabtu. Panjang satu kromosom adalah gabungan gen yang berdasarkan jumlah dari seluruh mata kuliah dan kelas yang ditawarkan di semester aktif, yang mana itu merupakan data pengampu. Dalam satu gen berisi informasi waktu, hari, dan ruang untuk satu mata kuliah atau data pengampu. Gen dalam kromosom direpresentasikan dengan bilangan *integer* yang menunjukkan *primary key* dari data pengampu, data jam, data hari, dan ruang. Gambar 3 menunjukkan skema pengkodean kromosom.



Gambar 3 Skema Pengkodean Kromosom

2. Inisialisasi Populasi. Di dalam tahap ini, pengkodean kromosom telah dilakukan. Satu kromosom terdiri dari 4 gen, yang merepresentasikan mata kuliah dengan dosen, ruang berdasarkan kategori mata kuliah, waktu, dan hari sebagaimana Gambar 2. Dalam satu populasi terdiri dari 416 kromosom, dimana ukuran populasi ditentukan berdasarkan *input* yang diberikan. Untuk penelitian ini, ukuran populasi sesuai hasil dari pengujian yaitu 30, itu berarti bahwa jumlah dari semua kromosom awal dalam 30 populasi adalah  $30 \times 416 = 12480$  kromosom.

3. Fungsi *Fitness*. Di dalam penelitian ini fungsi *fitness* dapat dilihat di Persamaan (1). Pada Persamaan (1) bahwa BD mendefinisikan banyaknya pelanggaran ruang dan waktu, BK mendefinisikan banyaknya pelanggaran dosen, dan WD mendefinisikan banyaknya pelanggaran waktu tidak bersedia. Setiap pelanggaran dari kromosom dengan *hard constraints* yang telah ditentukan, nilai pelanggaran bertambah 1 dan dijumlahkan semua nilai *fitness* dari semua kromosom dalam satu populasi sehingga tiap populasi memiliki nilai *fitness*.

$$Fitness = \frac{1}{1+(\sum BD + \sum BK + \sum WD)} \quad (1)$$

4. Seleksi. Proses seleksi dilakukan dengan membandingkan nilai *fitness* setiap populasi berdasarkan jumlah nilai *fitness* dari seluruh kromosom dalam tiap-tiap populasi, kemudian dilakukan perangkingan dan ditentukan induk kromosom.
5. *Crossover* dan Mutasi. Tahap ini merupakan pembetulan beberapa kromosom yang tidak sesuai dengan *hard constraints*. Kromosom yang kurang dari parameter Probabilitas *Crossover* ( $P_c$ ) dan Probabilitas Mutasi ( $P_m$ ) mengalami pindah silang dan mutasi. Pindah silang telah dilakukan dengan pindah silang dua titik berdasarkan parameter  $P_c$  kemudian dilakukan mutasi berdasarkan parameter  $P_m$ .

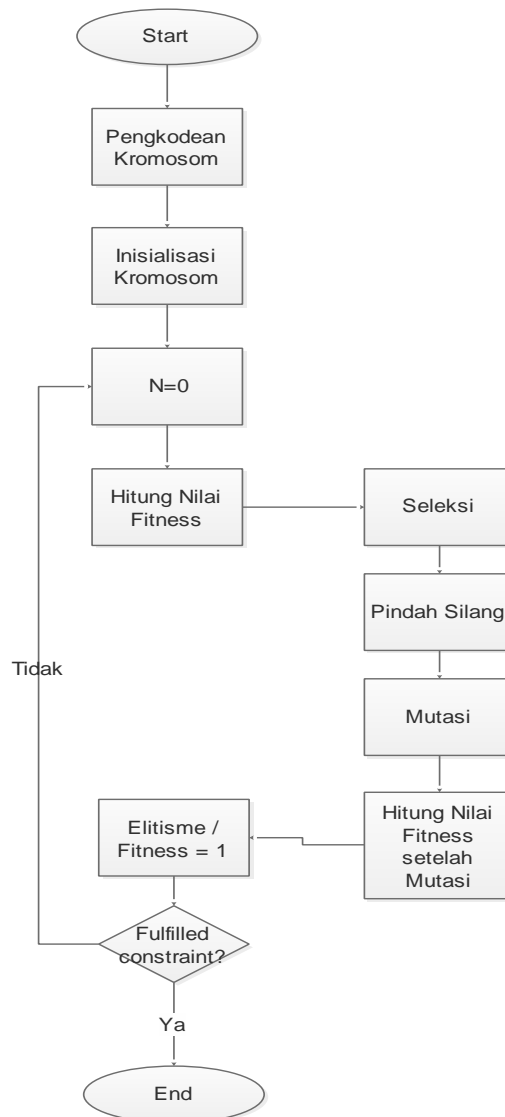
Langkah-langkah pindah silang dengan metode *two-points crossover* sebagai berikut:

1. Inisialisasi bilangan acak  $[0..1]$  dengan acuan parameter banyaknya populasi dan  $P_c$ .
2. Apabila bilangan acak yang dibangkitkan lebih kecil dari parameter  $P_c$ , kemudian dilakukan pemilihan kromosom secara acak sebagai induk. Jika lebih besar dari parameter  $P_c$  kembali ke poin 1.
3. Penentuan dua titik potong pindah silang.
4. Lakukan penyilangan di antara dua kromosom.
5. Kembali ke poin 1.
6. Berakhir.

Mutasi telah dilakukan dengan membangkitkan bilangan acak dan dibandingkan dengan parameter  $P_m$ . Seleksi kromosom yang akan dimutasi dilakukan secara acak. Setelah dilakukan mutasi, kromosom-kromosom dalam satu populasi dihitung kembali nilai *fitness*-nya. Jika nilai *fitness* dari tiap populasi tidak bernilai satu, ketika parameter banyaknya generasi mencapai sesuai yang ditentukan maka algoritma genetika berakhir, atau jika salah satu populasi setelah *crossover* dan mutasi bernilai satu maka algoritma genetika berakhir.

Langkah-langkah mutasi sebagai berikut:

1. Inisialisasi bilangan acak  $[0..1]$  dengan acuan parameter banyaknya populasi dan  $P_m$ .
2. Apabila bilangan acak yang dibangkitkan lebih kecil dari parameter  $P_m$ , kemudian ditentukan kromosom secara acak untuk dilakukan perubahan. Jika lebih besar dari parameter  $P_m$  maka kembali ke poin 1.
3. Hitung nilai *fitness* kembali untuk setiap kromosom dalam satu populasi.
4. Kembali ke poin 1.
5. Berakhir.



Gambar 4 Alur Algoritma

#### 4.2 Analisis Desain Tampilan

Beberapa data yang dibutuhkan dalam membangun penjadwalan mata kuliah, yaitu data perkuliahan tahun ajaran 2017-2018 semester genap yang terdiri dari data dosen, data jurusan, data kelas, data mata kuliah untuk tiap jurusan, data jam, data hari, data ruang, data waktu tidak bersedia dosen, dan data pengampu. Data-data tersebut diperoleh ketika dilakukannya wawancara dengan pihak akademik STT Wastukencana Purwakarta. Untuk proses penjadwalan atau terbentuknya data jadwal dengan algoritma genetika yang terdiri dari data pengampu, data jam, data ruang, dan data hari, kemudian dikodekan ke dalam kromosom.

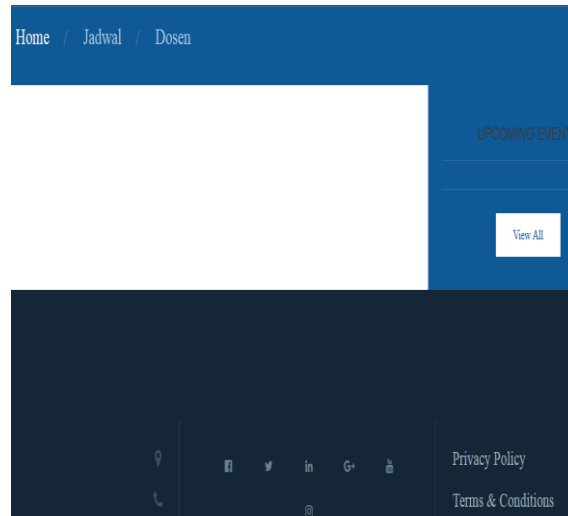
Untuk tampilan di dalam sistem menyediakan dua tampilan, yaitu tampilan depan (*front-end interface*) dan tampilan belakang (*back-end interface*). Tampilan depan diperuntukkan untuk admin, dosen, dan mahasiswa dalam mendapatkan informasi jadwal kuliah sedangkan tampilan belakang diperuntukkan untuk admin dalam mengelola data penjadwalan. Untuk mencetak jadwal, jadwal kuliah dikemas ke dalam *pdf file type*.

#### 5. PENGUJIAN

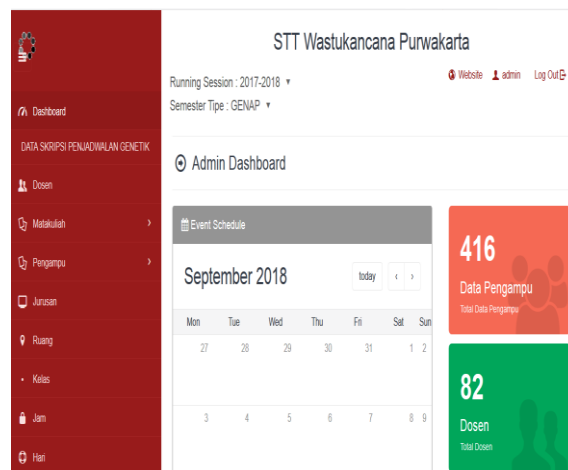
Implementasi dari analisis adalah sebuah sistem penjadwalan kuliah dengan algoritma genetika. Pengujian algoritma genetika mengacu kepada pencarian parameter algoritma genetika dengan dilakukannya beberapa simulasi sampai ditemukannya parameter tersebut agar dijadikan *default setting* untuk otomatis isian dari *form* buat jadwal. Untuk mengetahui berhasil atau tidak algoritma genetika sesuai parameter algoritma genetika yang dimasukkan, yaitu dengan notifikasi "Tidak Ada



Solusi Yang Optimal” apabila tidak berhasil sedangkan ketika itu berhasil maka tampilan jadwal perkuliahan ditampilkan. Gambar 5 dan Gambar 6 menunjukkan tampilan depan dan belakang dari sistem penjadwalan mata kuliah.



Gambar 5 Tampilan Depan



Gambar 6 Tampilan Belakang

Di tampilan depan terdapat menu Jadwal yang berfungsi untuk melihat jadwal. Ketika menu Jadwal diklik maka akan menampilkan sebuah tabel jadwal kuliah. Di dalam tabel jadwal kuliah terdapat tombol *print* untuk mencetak jadwal ke dalam *pdf file type* sebagaimana Gambar 7. Ketika data pengampu, data hari, data jam, dan data ruang sudah dilakukan *input data* ke dalam sistem, maka klik menu Penjadwalan di tampilan belakang di dalam kolom navigasi kiri sebagaimana Gambar 8 dan Gambar 9 yang menunjukkan tampilan jadwal kuliah yang sudah dibuat apabila algoritma genetika berhasil memproses pembuatan jadwal.

## Penjadwalan Mata Kuliah

| # | Hari  | Jam         | Dosen                     | Matakuliah           | SKS | Semester | Kelas  | Ruang    | Tahun Ajaran |
|---|-------|-------------|---------------------------|----------------------|-----|----------|--------|----------|--------------|
| 1 | Kamis | 20.00-20.20 | IMAY KURNIAWAN, M.Kom     | ORGANSASI KOMPUTER   | 3   | II       | IF/IMA | Ruang 10 | 2017-2018    |
| 2 | Kamis | 22.00-22.20 | MARKUM, M.Pd              | BAHASA INGGRES II    | 3   | II       | IF/IMA | Ruang 28 | 2017-2018    |
| 3 | Sabtu | 22.20-22.40 | RANI SRI WIRHYUNI, M.Hum  | BAHASA INDONESIA     | 3   | II       | IF/IMA | Ruang 12 | 2017-2018    |
| 4 | Sabtu | 18.40-19.00 | LISE SRI ANDAR MUNI, S.T. | MATEMATIKA DISKRIT 2 | 3   | II       | IF/IMA | Ruang    | 2017-2018    |

Gambar 7 Tabel Jadwal Kuliah

STT Wastukencana Purwokarta

Running Session : 2017-2018  
Semester Tipe : GENAP

Manage Penjadwalan

Buat Jadwal | List Jadwal | Loaded in : 2.6441 seconds

Jumlah Populasi:

Probabilitas CrossOver:

Probabilitas Mutasi:

Jumlah Generasi:

Gambar 8 Form Buat Jadwal

STT Wastukencana Purwokarta

Manage Penjadwalan

Buat Jadwal | List Jadwal | Loaded in : 2.6441 seconds

10 per page | Excel | PDF | Print | Search:

| # | Hari  | Jam               | Dosen                           | Matakuliah           | SKS | Semester | Kelas  | Ruang    |
|---|-------|-------------------|---------------------------------|----------------------|-----|----------|--------|----------|
| 1 | Senin | 21.20.00-22.40.00 | IRMA MEGAWATI M.Ag              | PENDIDIKAN AGAMA II  | 2   | II       | IF/IMA | Ruang 20 |
| 2 | Senin | 19.00.00-21.00.00 | IMAY KURNIAWAN, M.Kom           | ORGANSASI KOMPUTER   | 3   | II       | IF/IMA | Ruang 14 |
| 3 | Senin | 21.20.00-23.20.00 | MARKUM, M.Pd                    | BAHASA INGGRES II    | 3   | II       | IF/IMA | Ruang 30 |
| 4 | Senin | 11.20.00-13.20.00 | LISE SRI ANDAR MUNI, S.T., M.T. | MATEMATIKA DISKRIT 2 | 3   | II       | IF/IMA | Ruang 10 |

Gambar 9 Tampilan Jadwal Perkuliahan Tahun Ajaran 2017-2018

Beberapa simulasi dilakukan dalam pengujian algoritma genetika untuk mendapatkan parameter algoritma genetika yang menghasilkan solusi yang optimal, yaitu banyaknya populasi, nilai  $P_c$ , nilai  $P_m$ , dan banyaknya generasi atau iterasi maksimum. Solusi yang optimal disajikan ke dalam jadwal perkuliahan. Percobaan dilakukan tidak menggunakan data waktu tidak bersedia. Menurut [4] nilai  $P_m$  dan  $P_c$  diset umumnya antara 0 dan 1,  $P_c$  diset pada 0.5 dan  $P_m$  diset pada 0.37.

Pada simulasi pertama dilakukan dengan memberikan nilai  $P_c$  0.4 kurang 0.1 dari yang ditentukan menurut [4] dan  $P_m$  0.37. Tabel 4 menunjukkan hasil simulasi pertama.

Tabel 4 Simulasi 1

| No | Pm  | Pc   | Jumlah Populasi | Jumlah Generasi | Waktu (s) | Hasil |
|----|-----|------|-----------------|-----------------|-----------|-------|
| 1  | 0.4 | 0.37 | 5               | 10              | 12        | No    |
| 2  | 0.4 | 0.37 | 5               | 15              | 17        | No    |
| 3  | 0.4 | 0.37 | 5               | 20              | 22        | No    |
| 4  | 0.4 | 0.37 | 5               | 25              | 28        | No    |
| 5  | 0.4 | 0.37 | 5               | 50              | 55        | No    |
| 6  | 0.4 | 0.37 | 5               | 100             | 108       | No    |
| 7  | 0.4 | 0.37 | 5               | 300             | 326       | No    |
| 8  | 0.4 | 0.37 | 5               | 500             | 553       | No    |
| 9  | 0.4 | 0.37 | 5               | 1000            | 341/496   | yes   |
| 10 | 0.4 | 0.37 | 10              | 100             | 206       | No    |
| 11 | 0.4 | 0.37 | 10              | 300             | 433       | yes   |
| 12 | 0.4 | 0.37 | 10              | 500             | 462       | yes   |
| 13 | 0.4 | 0.37 | 10              | 1000            | 546       | yes   |
| 14 | 0.4 | 0.37 | 20              | 200             | 267       | yes   |
| 15 | 0.4 | 0.37 | 30              | 150             | 260       | yes   |

Pada Tabel 4 menjelaskan bahwa solusi yang optimal diperoleh ketika jumlah populasi 5 dan jumlah generasi 1000, jumlah populasi 10 dan jumlah generasi 300, jumlah populasi 10 dan jumlah generasi 500, jumlah populasi 10 dan jumlah generasi 1000, jumlah populasi 20 dan jumlah generasi 200, dan jumlah populasi 30 dan jumlah generasi 150. Waktu tercepat dalam simulasi pertama adalah 260 seconds ketika jumlah populasi 30 dan jumlah generasi 150.

Simulasi kedua dilakukan dengan memberikan nilai Pc 0.5 dan Pm 0.37. Tabel 5 menunjukkan hasil simulasi kedua.

Tabel 5 Simulasi 2

| No | Pm  | Pc   | Jumlah Populasi | Jumlah Generasi | Waktu (s) | Hasil |
|----|-----|------|-----------------|-----------------|-----------|-------|
| 16 | 0.5 | 0.37 | 5               | 10              | 12        | No    |
| 17 | 0.5 | 0.37 | 5               | 15              | 18        | No    |
| 18 | 0.5 | 0.37 | 5               | 20              | 23        | No    |
| 19 | 0.5 | 0.37 | 5               | 25              | 29        | No    |
| 20 | 0.5 | 0.37 | 5               | 50              | 55        | No    |
| 21 | 0.5 | 0.37 | 5               | 100             | 109       | No    |
| 22 | 0.5 | 0.37 | 5               | 300             | 322       | No    |
| 23 | 0.5 | 0.37 | 5               | 500             | 416       | yes   |
| 24 | 0.5 | 0.37 | 5               | 1000            | 1058      | yes   |
| 25 | 0.5 | 0.37 | 10              | 100             | 213       | No    |
| 26 | 0.5 | 0.37 | 10              | 300             | 563       | yes   |
| 27 | 0.5 | 0.37 | 10              | 500             | 639       | yes   |
| 28 | 0.5 | 0.37 | 10              | 1000            | 463       | yes   |

Pada Tabel 5 menjelaskan bahwa solusi yang optimal diperoleh ketika jumlah populasi 5 dan jumlah generasi 500, jumlah populasi 5 dan jumlah generasi 1000, jumlah populasi 10 dan jumlah generasi 300, jumlah populasi 10 dan jumlah generasi 500, dan jumlah populasi 10 dan jumlah generasi 1000. Waktu tercepat pada simulasi kedua adalah 416 seconds ketika jumlah populasi 5 dan jumlah generasi 500.

Simulasi ketiga dilakukan dengan memberikan nilai  $P_c$  0.73 dan  $P_m$  0.37. Tabel 6 menunjukkan hasil simulasi ketiga.

Tabel 6 Simulasi 3

| No | $P_m$ | $P_c$ | Jumlah Populasi | Jumlah Generasi | Waktu (s) | Hasil |
|----|-------|-------|-----------------|-----------------|-----------|-------|
| 29 | 0.73  | 0.37  | 5               | 10              | 22        | No    |
| 30 | 0.73  | 0.37  | 5               | 15              | 35        | No    |
| 31 | 0.73  | 0.37  | 5               | 20              | 44        | No    |
| 32 | 0.73  | 0.37  | 5               | 25              | 55        | No    |
| 33 | 0.73  | 0.37  | 5               | 50              | 109       | No    |
| 34 | 0.73  | 0.37  | 5               | 100             | 219       | No    |
| 35 | 0.73  | 0.37  | 5               | 500             | 654       | No    |
| 36 | 0.73  | 0.37  | 5               | 1000            | 903       | yes   |
| 37 | 0.73  | 0.37  | 10              | 50              | 110       | No    |
| 38 | 0.73  | 0.37  | 10              | 100             | 218       | No    |
| 39 | 0.73  | 0.37  | 10              | 500             | 457       | yes   |
| 40 | 0.73  | 0.37  | 10              | 1000            | 666       | yes   |

Pada Tabel 6 menjelaskan bahwa solusi yang optimal ketika jumlah populasi 5 dan jumlah generasi 1000, jumlah populasi 10 dan jumlah generasi 500, dan jumlah populasi 10 dan jumlah generasi 1000. Waktu tercepat pada simulasi ketiga adalah 457 seconds ketika jumlah populasi 10 dan jumlah generasi 500.

Dari ketiga simulasi yang dilakukan menunjukkan bahwa ketika banyaknya populasi dinaikkan dan banyaknya generasi pun juga sama, cenderung hasil dari algoritma genetika dalam memproses pembuatan jadwal menghasilkan solusi yang optimal. Waktu tercepat diraih dalam menghasilkan jadwal yang optimal yaitu ketika banyaknya populasi dinaikkan dan banyaknya generasi diperkecil. Hal itu menunjukkan semakin banyaknya populasi, maka banyaknya kromosom yang terbentuk dan waktu proses *looping* tidak membutuhkan waktu yang lama seiring banyaknya generasi yang ditentukan tidak terlalu besar.

## 6. PENUTUP

Berdasarkan hasil yang diperoleh bahwa:

1. Algoritma genetika dapat diterapkan di STT Wastukencana Purwakarta dalam pembuatan jadwal perkuliahan
2. Ketika  $P_c$  0.4,  $P_m$  0.37, jumlah populasi 30, dan jumlah generasi 150 menghasilkan solusi yang optimal dengan waktu tercepat yang dilakukan pada simulasi pertama. Hal ini menunjukkan bahwa parameter algoritma genetika diraih.
3. Untuk pengembangan algoritma genetika, diharapkan percobaan di penelitian selanjutnya menggunakan data waktu tidak bersedia, dan menjelaskan hasil dari setiap *looping*.

## DAFTAR PUSTAKA

- [1] E. Herjanto, *Manajemen Operasi*, Ketiga. Jakarta: Grasindo, 2007.
- [2] J. E. S. Eiben, Agoston E., *Introduction to Evolutionary Computing*, vol. 53. Heidelberg: Springer, 2003.

- [3] T. Widodo, *Komputasi Evolusioner Algoritma Genetik Pemrograman Genetik dan Pemrograman Evolusioner*. Yogyakarta: Graha, 2012.
- [4] S. H. R. L. Haupt, *Practical Genetic Algorithm*. Canada: John Willer & Sons. Inc., 2004.
- [5] Suyanto, *Algoritma Genetik dalam Matlab*. Yogyakarta: Andi Offset, 2005.
- [6] S. Kusumadewi, *Artificial Intelligence (Teknik & Aplikasinya)*. Yogyakarta: Graha Ilmu, 2003.
- [7] W. Hardyanto, "Applying an MVC Framework for The System Development Life Cycle with Waterfall Method Extended," *J.Phys Conf.*, no. Ser. 824012007, 2017.