# APLIKASI ALGORITMA GENETIK UNTUK OPTIMASI PENJADWALAN KEGIATAN PERKULIAHAN

*(Genetic Algorithm APllication for Optimization of Learning Activities Schedule)*

Robby Kurniawan Budhi
Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang

**Abstract**

*Scheduling learning activities in a campus is a complicated matter. The problem is often called the University Timetabling Problems (UTP). Beside the students factor, problems can also come from the lecturers. There are possibilities that the lecturers will teach more than one subject, because there is a possibility that the number of subjects and the number of lecturers is not comparable, so there must be a solution to figure that does not support two lecturers subjects on the same days and hours. In addition, it should also consider the availability of class so that learning activities can be carried out.*

*One method that can be used to complete the problem is to use the genetic algorithm approach. Genetic algorithm is a computational approach to solve the problem which use the process of biological evolution as the process.*

*It is expected that with the use of genetic algorithm will optimize the schedule so there will be no crash between the students' schedule, lecturers' schedule, and the availability of the class.*

***Keywords:*** *scheduling, optimization, genetic algorithms.*

## 1. PENDAHULUAN

Penjadwalan mata kuliah adalah masalah yang harus dihadapi di setiap pergantian semester dalam setiap tahun akademik. Tidak terkecuali di Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang (FTIK USM). Masalah penjadwalan ini menjadi hal yang memusingkan pihak Tata Usaha (TU) karena saat ini proses dilakukan secara manual dengan mempertimbangkan banyaknya prioritas ataupun kondisi perkuliahan.

Pada awalnya pihak TU akan mendata mata kuliah yang akan ditawarkan pada semester tersebut. Kemudian dilakukan pencatatan prioritas – prioritas terhadap kelas mata kuliah tertentu jika ada. Baru kemudian dilakukan proses penjadwalan dan membuat laporan hasil jadwal dengan menggunakan Microsoft Excel.

Namun prioritas yang diberikan terhadap suatu kelas, bisa terdiri dari beberapa komponen. Misalnya seorang dosen yang meminta waktu dan ruang kuliah tertentu. Otomatis hal tersebut akan dipertimbangkan terlebih dahulu dibandingkan dengan mata kuliah lain yang tidak memerlukan kondisi tertentu.

Kondisi lain yang harus diperhatikan adalah beda kapasitas ruang perkuliahan yang digunakan. Selain itu juga beda jumlah mahasiswa tiap angkatan yang akan mengikuti perkuliahan. Adapun kondisi perkuliahan, dalam hal ini perkuliahan teori, tiap angkatan akan dimasukkan dalam satu kelas yang sama. Serta kondisi – kondisi lain yang harus diperhatikan saat menyusun jadwal perkuliahan.

Hal lain yang harus diperhatikan adalah rasio dosen dan mata kuliah yang diajarkan. Hal ini mengingat jumlah dosen yang terbatas sedangkan mata kuliah yang diajarkan dalam satu semester meliputi beberapa angkatan mahasiswa sekaligus.

Dengan banyaknya prioritas serta kondisi yang harus diperhatikan tersebut, maka proses

penjadwalan mata kuliah menjadi hal yang membingungkan dan menggunakan waktu yang lama. Hal ini tentu saja menghambat kinerja TU yang masih harus menyelesaikan urusan administratif lain.

Secara khusus, penggunaan *game* jenis simulasi digunakan untuk mensimulasikan kejadian nyata yang dialami dalam melakukan proses penjadwalan mata kuliah. Hal ini sesuai dengan hakikat dari permainan simulasi yaitu untuk mewakili kondisi dunia nyata secara tepat (Wolf, 2001).

Sebuah penelitian oleh Yaman Barlas (2000) telah menghasilkan sebuah *game* yang mampu menganalisa dan menguji rencana strategis yang mampu dilakukan oleh perguruan tinggi berdasarkan data rasio jumlah mahasiswa, kualitas pengajar, serta tinggi rendahnya jumlah penelitian yang dilakukan. Hal ini membuktikan bahwa jenis *game* simulasi mampu digunakan untuk memberikan pertimbangan strategis dalam penentuan keputusan manajerial dalam sebuah institusi pendidikan.

Sedangkan metode yang digunakan untuk menyelesaikan masalah penjadwalan ini adalah menggunakan pendekatan algoritma genetika. Algoritma genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Algoritma ini biasa digunakan untuk proses optimasi dan pencarian hasil yang terbaik (Whitley, 1993).

Diharapkan dengan digunakannya algoritma genetika akan mempercepat proses penjadwalan mata kuliah yang memenuhi kondisi dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, tidak ada permasalahan bentrokan jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai secara fasilitas untuk seluruh mata kuliah yang ada.

## 2. PENGATURAN JADWAL KULIAH DI FTIK USM

Pengaturan jadwal kuliah di FTIK USM dilakukan dua kali dalam satu tahun ajaran yaitu pada semester gasal dan semester genap. Dalam mengatur jadwal, ada beberapa tahapan yang harus dijalankan oleh staf TU FTIK USM antara lain :

1. Pendataan kurikulum
   Pendataan ini dilakukan untuk menentukan mata kuliah apa saja yang ditawarkan pada semester yang akan berjalan beserta jumlah SKS untuk masing – masing mata kuliah. Adapun satuan waktu untuk satu SKS adalah satu jam.
2. Pengecekan jumlah mahasiswa
   Data jumlah mahasiswa aktif dalam masing – masing angkatan digunakan untuk menentukan kelas mana yang nantinya bisa digunakan untuk masing – masing angkatan tersebut, mengingat kapasitas ruang kuliah yang berbeda – beda.
3. Pengecekan jumlah kelas dan kapasitas ruang
   Dalam tiap semester, dilakukan pendataan ulang ruang yang bisa digunakan untuk perkuliahan beserta kapasitasnya. Hal ini mengingat adanya kemungkinan untuk meminjam ruang dari fakultas lain untuk perkuliahan.
4. Penentuan dosen pengampu untuk masing – masing mata kuliah yang ditawarkan
   Hal ini dilakukan untuk menentukan prioritas waktu serta mencegah bentrokan jadwal apabila seorang dosen mengampu lebih dari satu mata kuliah. Mengingat rasio antara dosen dan mata kuliah yang tidak berimbang, maka ada kemungkinan bahwa seorang dosen mampu mengajar beberapa mata kuliah untuk satu angkatan ataupun angkatan yang berbeda.
5. Penentuan waktu kuliah
   Adapun waktu kuliah yang bisa digunakan adalah pukul 08.00 – 15.00 untuk hari Senin hingga Kamis, dan pukul 08.00 – 12.00 untuk hari Jumat. Hal ini diperlukan mengingat beda jumlah SKS masing – masing mata kuliah yang ditawarkan.

Adapun unit waktu untuk satu SKS adalah satu jam.

6. Penentuan prioritas untuk mata kuliah atau dosen tertentu

   Prioritas ini digunakan karena adanya dosen luar yang diperbantukan sebagai dosen pengampu dan menginginkan waktu tertentu untuk jam mengajarnya. Demikian juga untuk dosen – dosen yang mengajar beberapa mata kuliah sekaligus untuk satu semester sehingga prioritas in iberguna untuk menghindari bentrokan jadwal dengan mata kuliah lainnya.

Dari sekian banyak proses tersebut, baru kemudian akan dibuat penjadwalan mata kuliah yang memperhatikan kondisi dan prioritas masing – masing.

## 3. ALGORITMA GENETIK

### 3.1 Pengertian Algoritma Genetik

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg. [1] Algoritma genetik 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. Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi.

Untuk menggunakan algoritma genetik, solusi permasalahan direpresentasikan sebagai khromosom. Tiga aspek yang penting untuk penggunaan algoritma genetik:

1. Definisi *fitness function*
2. Definisi dan implementasi representasi genetik
3. Definisi dan implementasi operasi genetik

Jika ketiga aspek di atas telah didefinisikan, algoritma genetik generik akan bekerja dengan baik.

Tentu saja, algoritma genetik bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk untuk mencari penyelesaian dari fungsi analitis *convex* yang "berperilaku baik" yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetik karena metode ini dengan cepat menemukan solusi minimum ketika algoritma genetik masih menganalisa bobot dari populasi awal. Untuk problem – problem ini pengguna harus mengakui fakta dari pengalaman ini dan memakai metode tradisional yang lebih cepat tersebut. Akan tetapi, banyak persoalan realistis yang berada di luar golongan ini. Selain itu, untuk persoalan yang tidak terlalu rumit, banyak cara yang lebih cepat dari algoritma genetik. Jumlah besar dari populasi solusi, yang merupakan keunggulan dari algoritma genetik, juga harus mengakui kekurangannya dalam kecepatan pada sekumpulan komputer yang dipasang secara seri –*fitness function* dari tiap solusi harus dievaluasi. Namun, bila tersedia komputer-komputer yang paralel, tiap prosesor dapat mengevaluasi fungsi yang terpisah pada saat yang bersamaan. Karena itulah, algoritma genetik sangat cocok untuk perhitungan yang paralel.

### 3.2 Teknik Penggunaan Algoritma Genetik

Algoritma genetik dimulai dengan sekumpulan set status yang dipilih secara random, yang disebut populasi. Algoritma ini mengkombinasikan dua populasi induk. Setiap status atau individual direpresentasikan sebagai sebuah string.

***Fitness function***

Setiap individual dievaluasi dengan *fitness function*. Sebuah *fitness function* mengembalikan nilai tertinggi untuk individual yang terbaik. Individu akan diurutkan

berdasarkan nilai atau disebut dengan selection.

***Crossover***

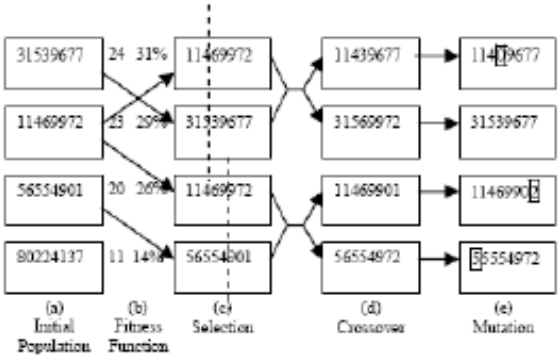
Untuk setiap pasang induk, sebuah titik *crossover* akan dipilih secara random dari posisi dalam string. Pada gambar titik *crossover* terletak pada indeks ketiga dalam pasangan pertama dan setelah indeks kelima pada pasangan kedua.

**Mutasi**

Pada mutasi, tiap lokasi menjadi sasaran mutasi acak, dengan probabilitas independen yang kecil. Sebuah digit dimutasikan pada anak pertama, ketiga, dan keempat. Algoritma genetik mengkombinasikan suatu kecenderungan menaik dengan pengeksplorasian acak di antara thread pencarian paralel. Keuntungan utamanya,bila ada, datang dari operasi *crossover*. Namun, secara matematis dapat tunjukkan bahwa bila posisi dari kode genetik di permutasikan di awal dengan urutan acak, *crossover* tidak memberikan keunggulan. Secara intuisi, keuntungannya didapat dari kemampuan *crossover* untuk menggabungkan blok-blok huruf berukuran besar yang telah berevolusi secara independen untuk melakukan fungsi yang bermanfaat sehingga dapat menaikkan tingkat granularity di mana pencarian dilakukan.

***Schema***

Teori dari algoritma genetik menjelaskan cara kerjanya menggunakan ide dari suatu *schema*, suatu substring di mana beberapa posisi tidak disebutkan. Dapat ditunjukkan bahwa, bila *fitness* rata-rata dari *schema* berada di bawah mean maka jumlah instansiasi dari *schema* di dalam populasi akan bertambah seiring bertambaahnya waktu. Jelas sekali bahwa efek ini tidak akan signifikan bila bit-bit yang bersebelahan sama sekali tidak berhubungan satu sama sekali, karena akan ada beberapa blok kontigu yang memberikan keuntungan yang konsisten. Algoritma genetik paling efektif dipakai bila *schema-schema* berkorespondensi menjadi komponen berati dari sebuah solusi. Sebagai contoh, bila string adalah representasi dari sebuah antena, maka *schema* merepresentasikan komponen-komponen dari antena, seperti reflector dan deflector. Sebuah komponen yang baik cenderung akan berkerja baik pada rancangan yang berbeda. Ini menunjukkan bahwa penggunaan algoritma genetik yang benar memerlukan rekayasa yang baik pada representasinya.




**Gambar 1. Contoh Penggunaan Algoritma Genetik**

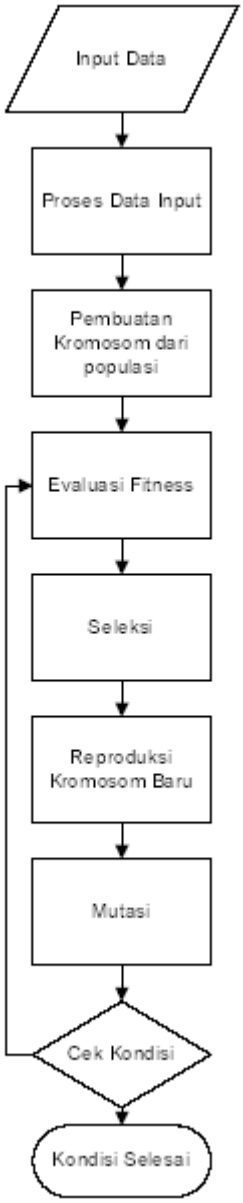
### 3.3 Pseudo Code Algoritma Genetik

**function** GenetikAlgorithm(population,*Fitness*-FN) **->** an individual
{input berupa population, sebuah kumpulan individual dan *Fitness*-FN, sebuah fungsi yang mengukur *fitness* suatu individual}
deklarasi
i,x,y : integer
**algoritma**
**repeat**
new_population**<-**empty set
**for** i=1 **to** size(population) **do**
x**<-**RandomSelection(population, *Fitness*-FN)
y**<-**RandomSelection(population, *Fitness*-FN)
child<-Reproduce(x,y)
**if**(smallRandomProbability) **then**
child**<-**mutate(child)
add child to new_population
population**<-**new_population
**until** some individual is fit enough or the time has elapsed
**return** the best individual in population(based on *Fitness*-FN)
**function** Reproduce(x,y : parent individuals)**->**individual
**deklarasi**
**algoritma**
n**<-**length(x)
c**<-**random number from 1 to n
**return** Append(substring(x,1,c), substring(y,c +1,n))

## 4. PENERAPAN ALGORITMA GENETIK PADA OPTIMASI PENJADWALAN

### 4.1 Flowchart Program

Flowchart program dapat dilihat pada Gambar 2. Flowchart ini terdiri dari delapan sub program yaitu input data, proses data input, pembuatan kromosom dan populasi, evaluasi *fitness*, seleksi, reproduksi kromosom baru, mutasi, serta kondisi selesai.




**Gambar 2. Flowchart Program**

Input, output dan proses dari setiap subprogram akan dibahas berikut ini.

**Input Data**

Terdapat enam masukan kelompok data yang perlu diberikan, yaitu :
1. Tabel Mata Kuliah
2. Tabel Dosen
3. Tabel Kelas
4. Tabel Ruang
5. Bobot *Fitness*
6. Kondisi Selesai

Tabel Mata Kuliah berisikan daftar seluruh mata kuliah yang akan dilaksanakan pada semester yang bersangkutan. Data yang perlu disertakan untuk setiap mata kuliah yang ada adalah kelas peserta, jumlah sks, dosen pengajar serta ruangan untuk mata kuliah tersebut.

Tabel Dosen, Tabel Kelas, serta Tabel Ruang adalah tabel waktu yang menginformasikan waktu-waktu dari dosen, kelas dan ruangan yang dapat digunakan untuk mata kuliah yang bersangkutan.

**Proses Data Input**

Agar dapat diproses dalam algoritma ini Tabel Mata Kuliah, Tabel Dosen, Tabel Kelas dan Tabel Ruang harus digabungkan terlebih dahulu menjadi Tabel Prioritas Mata Kuliah. Untuk menjadwalkan suatu Mata Kuliah, perlu mempertimbangkan jadwal waktu dosen, kelas dan ruangan yang tersedia. Maka setiap mata kuliah akan memiliki banyaknya pilihan penjadwalan yang berbeda. Bisa jadi ada mata kuliah yang memiliki tiga pilihan hari dan bisa jadi ada mata kuliah yang hanya memiliki satu pilihan hari saja. Tabel Prioritas Mata Kuliah berisikan banyaknya tingkat pilihan penjadwalan dari setiap mata kuliah yang ada serta telah diurutkan dari mata kuliah yang paling sedikit pilihan penjadwalannya hingga mata kuliah yang terbanyak pilihan penjadwalannya. Dari proses ini diharapkan tidak ada mata kuliah yang tidak dapat teralokasikan penjadwalannya dikarenakan pada jadwal – jadwal yang memungkinkan bagi mata kuliah tersebut telah digunakan oleh mata kuliah lainnya.

**Pembuatan Kromosom dan Populasi**

Berdasarkan urutan dari Tabel Prioritas Mata Kuliah, setiap mata kuliah akan dijadwalkan ke dalam Tabel Jadwal Mata Kuliah secara acak. Agar diketahui apakah pada waktu tersebut dosen, kelas, maupun ruangan dapat digunakan untuk melaksanakan perkuliahan, maka Tabel Dosen, Tabel Ruang, dan Tabel Kelas untuk setiap mata kuliah serta Tabel Mata Kuliah harus dipetakan terlebih dahulu dalam Tabel Jadwal Mata Kuliah.

Sebelum menjadwalkan suatu mata kuliah pada Tabel Jadwal Mata Kuliah, algoritma akan mengecek terlebih dahulu kepada Tabel Jadwal Mata Kuliah Bayangan untuk mengetahui apakah pada waktu tersebut dapat digunakan untuk perkuliahan atau tidak. Jika tidak maka algoritma didesain untuk mencari alokasi waktu lainnya.

**Evaluasi *Fitness***

Faktor-faktor yang mempengaruhi evaluasi *fitness* terhadap alternatif solusi adalah sebagai berikut :

1. Pemecahan mata kuliah; Terhadap mata kuliah dengan bobot 3 SKS, program dapat memecah mata kuliah tersebut menjadi 2 atau 3 kelompok jam kuliah jika waktu penjadwalan yang ada tidak memungkinkan untuk dilaksanakannya mata kuliah tersebut dalam satu waktu. Hal ini dibuat dengan tujuan memperluas kemungkinan alternatif penjadwalan yang ada terutama pada mata kuliah yang hanya memiliki sedikit alternatif penjadwalan. Tetapi pemecahan mata kuliah ini akan memperkecil nilai *Fitness*, sehingga kelak program akan cenderung menyeleksi solusi penjadwalan yang memiliki pemecahan mata kuliah yang terlalu banyak.
2. Pemadatan di suatu waktu; Untuk meningkatkan produktivitas pemakaian ruangan, maka dikehendaki agar ruangan dapat segera digunakan semenjak pagi hari. Jika program menawarkan solusi penjadwalan di mana terdapat waktu pagi yang tidak digunakan, maka hal ini akan memperkecil nilai *fitness* solusi.
3. Frekuensi mengajar dosen; Diinginkan agar tugas mengajar dosen dapat terdistribusi merata di tiap hari kerjanya dengan tujuan agar performansi dosen sewaktu mengajar dapat tetap dijaga optimal. Nilai *fitness* solusi akan berkurang jika dalam solusi tersebut terdapat dosen yang memiliki tingkat mengajar terlalu tinggi dalam satu harinya.
4. Frekuensi belajar mahasiswa; Seperti halnya pada dosen, untuk menjaga performansi belajar mahasiswa maka diharapkan tidak ada jadwal kuliah yang terlalu padat dalam satu hari. Jika solusi menawarkan jadwal kuliah kelas yang terlalu padat dalam satu hari, maka nilai *fitness* solusi yang berkurang. Kelas yang memiliki kuliah lebih dari 5 sks pada satu hari didefinisikan sebagai kelas yang memiliki frekuensi kuliah yang tinggi.
5. Kedekatan antar mata kuliah; Idealnya mahasiswa memiliki waktu istirahat antar dua mata kuliah yang ada dalam satu hari sehingga kelelahan mahasiswa dalam mengikuti mata kuliah pertama tidak mengganggu proses belajar pada

mata kuliah selanjutnya. Didefinisikan bahwa dua mata kuliah yang berjarak kurang dari dua sks untuk satu kelasnya digolongkan sebagai mata kuliah yang berdekatan dan dapat memperkecil nilai *fitness* dari solusi yang ditawarkan program. Walaupun ada waktu jeda antar dua mata kuliah, tetapi diharapkan juga mahasiswa tidak menunggu terlalu lama antar dua mata kuliah tersebut.

Rumus *fitness* yang digunakan adalah sebagai berikut:

$$Fitness = \frac{1}{B1xF1 + B2xF2 + B3xF3 + B4xF4 + B5xF5 + B6xF6 \ldots}$$

dengan :

F1 = Banyaknya mata kuliah yang dipecah
F2 = Banyaknya waktu pagi yang kosong
F3 = Banyaknya frekuensi jam mengajar yang tinggi dari seorang dosen
F4 = Banyaknya frekuensi jam kuliah yang tinggi dari satu kelas
F5 = Banyaknya mata kuliah yang berdekatan
F6 = Banyaknya mata kuliah yang berjauhan.
B1 = Bobot pemecahan mata kuliah
B2 = Bobot waktu pagi yang kosong
B3 = Bobot frekuensi mengajar dosen
B4 = Bobot frekuensi kuliah kelas
B5 = Bobot mata kuliah yang berdekatan
B6 = Bobot mata kuliah yang berjauhan

Setiap faktor yang mempengaruhi nilai *fitness* di atas memiliki tingkat pengaruh yang berbeda terhadap nilai *fitness*. Tingkat pengaruh ini disebut sebagai bobot. Jika suatu faktor pengaruh memiliki harga bobot yang tinggi maka setiap kali faktor tersebut terjadi dalam suatu solusi maka akan sangat mengurangi nilai *fitness* dari solusi tersebut. Dan sebaliknya jika suatu faktor memiliki harga bobot yang kecil, maka tidak akan terlalu mengurangi nilai *fitness* dari solusi meskipun faktor tersebut banyak terjadi dalam solusi yang ditawarkan.

Dari rumus nilai *fitness* di atas dapat terlihat bahwa yang mempengaruhi besar nilai *fitness* adalah harga FN karena harga BN akan tetap selama proses. Jika harga FN semakin besar maka nilai *Fitness* akan semakin kecil. Karena diinginkan solusi yang memiliki nilai *Fitness* yang besar, maka program ini diharapkan tidak terlalu banyak memunculkan faktorfaktor pengaruh ini dalam solusi yang ditawarkan.

**Seleksi**

Untuk mendapatkan solusi yang terbaik, maka program harus menyeleksi solusi yang memiliki nilai *fitness* yang tergolong rendah. Seleksi menggunakan metode good *fitness* yaitu setengah dari jumlah populasi yang memiliki harga *fitness* yang terendah akan dihilangkan sehingga akan hanya selalu tersisa sekelompok solusi yang terbaik yang pernah diperoleh oleh program. Solusi yang tersisa hasil seleksi ini dikenal dengan nama populasi induk.

Agar jumlah populasi tetap, maka perlu dibangkitkan solusi baru sebanyak setengah dari jumlah populasi yang ada. Dalam program ini, cara yang digunakan untuk membangkitkan solusi baru menggunakan dua cara yaitu reproduksi kromosom baru dan cara mutasi dari solusi induk. Tujuan pembangkitan solusi baru ini untuk menemukan alternatif solusi yang lebih baik dari solusi – solusi yang sudah diperoleh.

**Reproduksi Kromosom Baru**

Setengah dari jumlah populasi baru akan dibangkitkan dengan cara reproduksi kromosom baru. Yaitu penyusunan alternatif solusi penjadwalan secara acak kembali untuk setiap mata kuliah. Proses ini sama dengan langkah ketiga yang telah dibahas. Dengan proses ini maka akan dihasilkan sekelompok populasi baru yang benar-benar berbeda dengan populasi induknya.

**Mutasi**

Adapun setengah populasi baru lainnya akan dibangkitkan dengan cara mutasi. Yaitu setengah dari populasi induk akan dipilih untuk diduplikasi. Pemilihan dapat dilakukan dengan metode good *fitness*, random maupun roulette whell. Pada hasil duplikasi ini akan dilakukan sedikit percobaan terhadap posisi penjadwalan beberapa mata kuliah. Proses mutasi ini adalah suatu proses eksploitasi terhadap kemungkinan – kemungkinan modifikasi pada jadwal yang telah ada. Perubahan posisi beberapa mata kuliah ini (mutasi) dapat membuat solusi duplikasi ini 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 bisa jadi pada iterasi berikutnya diperoleh solusi hasil mutasi yang lebih baik nilai *fitness*nya daripada solusi induknya. Maksudnya adalah tidak menjadi masalah jika solusi hasil mutasi ini mengalami penurunan nilai *fitness* daripada solusi induknya. Kita hanya perlu untuk mengacuhkannya saja dan tetap memberikan perhatian pada solusi terbaik yang telah dicapai.

Pada proses mutasi ini, program akan memilih empat mata kuliah secara acak untuk dikeluarkan dari solusi hasil penggandaan tersebut untuk dijadwalkan kembali pada posisi yang berbeda. Sebelum mata kuliah yang dikeluarkan tersebut dijadwalkan kembali, maka program harus memproses data input terlebih dahulu dari keempat mata kuliah tersebut. Proses ini sama dengan proses pada langkah kedua dari algoritma pemrograman ini. Hal ini dilakukan untuk menghindari mata kuliah yang memiliki pilihan penjadwalan yang sedikit menjadi tidak bisa diposisikan kembali dalam penjadwalan karena pilihan-pilihan waktu penjadwalannya telah terisi oleh mata kuliah yang dikeluarkan lainnya.

**Kondisi Selesai**

Terdapat tiga kondisi selesai yang dapat menghentikan proses algoritma pemrograman ini, yaitu:

1. Jika setelah beberapa generasi berturut-turut nilai *fitness* terbaik dari populasi tidak mengalami perubahan kembali
2. Jika jumlah generasi atau iterasi maksimum telah tercapai.
3. Jika nilai *fitness* terbaik minimal telah tercapai.

Jika salah satu kondisi di atas telah diperoleh maka iterasi akan dihentikan.dan jika salah satu kondisi selesai ini belum tercapai maka program akan mengulang kembali proses ini / iterasi dari langkah keempat yaitu evaluasi *fitness* terhadap populasi baru tadi.

## 5. KESIMPULAN

Dengan bantuan Algoritma Genetik penyusunan penjadwalan mata kuliah dapat dioptimalkan. Program dapat mencari solusi penjadwalan pada waktu yang dapat digunakan baik oleh dosen, kelas maupun ruangan yang terlibat dalam suatu mata kuliah. Di samping itu, program dapat meminimalkan tingginya frekuensi mengajar seorang dosen, frekuensi kuliah suatu kelas dan faktor pengaruh lainnya.

Proses penjadwalan mata kuliah menggunakan Algoritma Genetik ini dapat diterapkan pada kasus-kasus penjadwalan dengan multi angkatan dan multi ruangan. Dengan menggunakan metode *best fitness*, maka Algoritma Genetik akan selalu menunjukkan kenaikan *fitness* atau dengan kata lain generasi selanjutnya lebih baik atau minimal sama dengan generasi sebelumnya.

## DAFTAR PUSTAKA


[1] Randy L. Haupt . 2004. "*Practical Genetic Algorithms*". A John Wiley & Sons, Inc.

[2] Muhammad Aria. 2006. "*Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Mata Kuliah*". Universitas Komputer Indonesia.

[3] Wilhelm Erben 2005 "A Hybrid Grouping Genetic Algorithm for Examination Timetabling". University of
Applied Sciences.

[4] http://www.esat.kuleuven.ac.be/sista/mai/2006-2007/cs/gafortimetabling.pdf Tanggal Akses 20 Mei 2008

[5] http://ilmukomputer.com/2007/03/29/algoritma-genetikadan-contoh-aplikasinya/ . Tanggal Akses 20 Mei 2008

[6] http://lancet.mit.edu/~mbwall/presentations/IntroTo GAs/ .Tanggal akses 20 Mei 2008