

KONSEP ALGORITMA GENETIK BINER UNTUK OPTIMASI PERENCANAAN JADWAL KEGIATAN PERKULIAHAN

(Binary Genetic Algorithm Concept to Optimize Course Timetabling)

Iwan Aang Soenandi

Fakultas Teknik dan Ilmu Komputer Jurusan Teknik Industri
Universitas Kristen Krida Wacana
Jl. Tanjung Duren Raya No. 4, Jakarta Barat 11470
iwan.as@ukrida.ac.id

Abstrak

Kegiatan penjadwalan perkuliahan di suatu kampus bukanlah hal yang sederhana. Banyak faktor-faktor yang harus dijadikan sebagai pertimbangan dalam penyusunan penjadwalan ini. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan pendekatan algoritma genetik. Algoritma genetik merupakan pendekatan komputasional untuk menyelesaikan masalah, yang dimodelkan dengan proses biologi dari evolusi. Dengan digunakannya algoritma genetik biner diharapkan akan diperoleh optimasi penjadwalan, yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan dengan tahapan perhitungan nilai *fitness* terbaik.

Kata Kunci: penjadwalan, optimasi, algoritma genetik

Abstract

Course timetabling in a university is a complex activity that requires many factors to consider. One method that can be used to solve the timetabling problems is by using genetic algorithm approach. Genetic algorithm is a computational approach employed to solve problems as modeled by the biological evolution process. The use of binary genetic algorithm is expected to optimize course timetabling in which best combination of courses and lecturers occur by calculating the fitness value.

Keywords: scheduling, optimization, genetic algorithm

Tanggal Terima Naskah : 10 Juni 2013

Tanggal Persetujuan Naskah : 19 Juni 2013

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Aktivitas penjadwalan perkuliahan di suatu perguruan tinggi merupakan hal yang tidak sederhana. Terdapat berbagai aspek yang berkaitan dalam proses penjadwalan tersebut yang harus diperhatikan, diantaranya:

- 1) Tidak semua waktu dapat digunakan oleh dosen untuk mengajar.
- 2) Tidak boleh adanya jadwal kuliah yang bentrok dengan jadwal kuliah yang lain.
- 3) Distribusi jadwal perkuliahan diharapkan dapat merata untuk setiap kelas per hari.
- 4) Adanya mata kuliah yang sudah dijadwalkan sebelumnya.

- 5) Terdapat perbedaan sks atau lama waktu perkuliahan.
- 6) Perencanaan jadwal mata kuliah ini akan semakin rumit jika terdapat banyak kelas paralel yang dibuka pada setiap angkatannya.

Hal lain yang sering terjadi adalah dalam perencanaan penyusunan jadwal kuliah ini terdapat sangat banyak kemungkinan, yang selayaknya dicoba untuk menemukan penjadwalan yang terbaik. Oleh karena itu, dibutuhkan metode optimasi yang dapat diterapkan untuk mengerjakan penjadwalan mata kuliah ini, salah satunya yang akan digunakan adalah Metode Algoritma Genetik Biner.

1.2 Batasan Masalah

Agar konsep penelitian ini lebih terarah, perlu ditentukan batasan permasalahan. Adapun batasan yang digunakan dalam penelitian ini, yaitu:

- 1) Setiap dosen bersedia mengajar di ruang manapun yang tersedia dan pada waktu yang ditentukan.
- 2) Setiap mahasiswa bersedia menempati ruang dan waktu yang tersedia.
- 3) Kapasitas ruangan pasti mencukupi sesuai dengan jumlah mahasiswa.
- 4) Metode pemecahan masalah dengan algoritma genetika.
- 5) Data lain untuk melengkapi algoritma dapat diasumsikan.

2. ALGORITMA GENETIK

2.1 Konsep Dasar Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antarindividu organisme [1]. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup. Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin, yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol.

Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak, terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk dengan menggunakan operasi penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak, serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, algoritma ini akan konvergen ke kromosom terbaik [2].

2.2 Metode Seleksi dengan Roda *Roulette*

Metode *Roulette Wheel* merupakan salah satu metoda seleksi yang banyak dipergunakan. *Roulette wheel* menyeleksi populasi baru dengan distribusi probabilitas yang berdasarkan pada nilai kepantasan (*fitness*) [3]. Proses seleksi dilakukan dengan cara membuat kromosom yang mempunyai fungsi objektif terkecil/terbesar memiliki kemungkinan terpilih yang besar atau mempunyai nilai probabilitas yang tinggi.

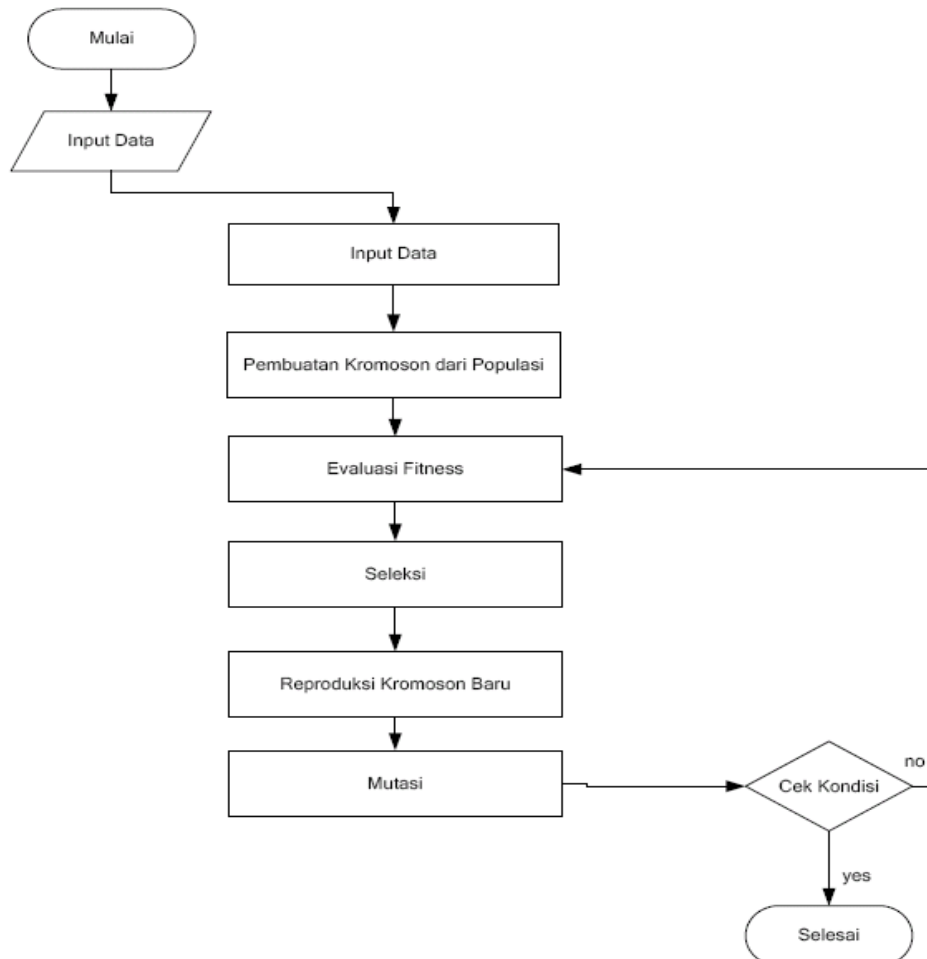
2.3 Crossover

Tahapan *Crossover* (penyilangan) dilakukan atas dua kromosom untuk menghasilkan kromosom anak (*offspring*). Kromosom anak yang terbentuk akan mewarisi sebagian sifat kromosom induknya. Metode *crossover* yang akan digunakan untuk kasus ini adalah Metode *Order (OX)*. Pada metode ini, *offspring* dibentuk dengan cara memilih sebagian jalur dari suatu induk, kemudian menata ulang jalur berdasarkan urutan tertentu dari induk yang lainnya dengan urutan yang sama dari induk yang lain [3].

3. PENERAPAN ALGORITMA GENETIK PADA OPTIMASI PERENCANAAN PENJADWALAN

3.1 Flowchart Program

Flowchart program ditunjukkan pada Gambar 1. *Flowchart* ini terdiri atas delapan sub program, yaitu *input* data, proses data *input*, pembuatan kromosom dan populasi, evaluasi *fitness*, seleksi, reproduksi kromosom baru, mutasi, cek kondisi, serta selesai.



Gambar 1. *Flowchart*

3.2 Pseudo Code Algoritma Genetik

Pseudo Code untuk Algoritma genetik adalah sebagai berikut.

```
function
AlgoritmaGenetik(populasi)
{Bangkitkan populasi, sebuah kumpulan individual, jumlah iterasi maksimum}
deklarasi
set binary code for populasi
i,x,y : integer
iterasi t=1
algoritma elitisme
populasi baru<-set kosong
set kendala khusus {mata kuliah pada jam tertentu,dll}
for i=1 to size(populasi) do
x<-RandomSelection(populasi)
y<-RandomSelection(populasi)
anak<-Reproduksi(x,y)
if(smallRandomProbability) then
anak<-mutasi(anak)
tambah anak to populasi baru
populasi<-populasi baru
until individu sudah sesuai
or waktu sudah terlampaui or jumlah iterasi sudah cukup
return individu terbaik in populasi (based on Fitness-FN)
function Reproduksi(x,y : individu orang tua)
test Fitness-FN
iterasi t=t+1
repeat
n<-length(x)
c<-bilangan random dari 1 sampai n
return
Delay
substring(x,1,c),
substring(y,c +1,n))
End
```

3.3 Pelaksanaan Program

3.3.1 Input Data

Terdapat enam masukan kelompok data yang perlu diberikan, yaitu Data Mata Kuliah, Data Dosen, Data Kelas, Data Ruang, Bobot *Fitness*, dan 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.

3.3.2 Proses Data Input

Agar dapat diproses dalam algoritma ini semua data tersebut dibuat dalam bentuk tabel sehingga akan terbentuk beberapa tabel, yaitu 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. Karena itu, setiap mata kuliah akan memiliki jumlah pilihan penjadwalan yang berbeda, misalnya terdapat mata kuliah yang memiliki tiga pilihan hari dan terdapat mata kuliah yang hanya memiliki satu pilihan hari saja.

Tabel 1. Rencana *data input*

	Mata Kuliah 1	Mata Kuliah 2	Mata Kuliah n
Dosen A			
Dosen B			
Dosen X			

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, termasuk mata kuliah yang sudah ditetapkan sebelumnya, misalnya mata kuliah MPK. Dari proses ini diharapkan tidak ada mata kuliah yang tidak teralokasikan penjadwalannya dikarenakan pada jadwal-jadwal yang memungkinkan bagi mata kuliah tersebut telah digunakan oleh mata kuliah lainnya.

3.3.3 Pembuatan Kromosom dan Populasi

Dalam perencanaan ini 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. Setelah itu, populasi dalam bentuk biner dibuat untuk diolah pada tahap berikutnya.

3.3.4 Evaluasi *Fitness*

Pada setiap iterasi, faktor-faktor yang mempengaruhi evaluasi *fitness* terhadap alternatif solusi adalah sebagai berikut:

- 1) Pemecahan setiap 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. Namun, pemecahan mata kuliah ini akan memperkecil nilai *Fitness*, sehingga kelak program akan cenderung menseleksi solusi penjadwalan yang memiliki pemecahan mata kuliah yang terlalu banyak.
- 2) Pengelompokan di suatu waktu; Untuk meningkatkan produktivitas pemakaian ruangan, maka dikehendaki agar ruangan dapat segera digunakan semenjak pagi hari. Jika program menawarkan solusi penjadwalan dimana terdapat waktu pagi yang tidak digunakan, maka hal ini akan memperkecil nilai *fitness* solusi.
- 3) Frekuensi belajar mahasiswa; Seperti halnya pada dosen, untuk menjaga performansi belajar mahasiswa, 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.

- 4) Jeda waktu kosong antarmata kuliah; para mahasiswa mengharapkan memiliki waktu istirahat antara 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 antara dua mata kuliah, tetapi diharapkan mahasiswa tidak menunggu terlalu lama antara dua mata kuliah tersebut.

Rumus *fitness* yang digunakan adalah sebagai berikut [4]:

$$F(x) = \frac{1}{1 + f(x)'} \dots\dots\dots(1)$$

dengan:

$f(x)$ = fungsi tujuan dari *problem*

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 akan sangat mengurangi nilai *fitness* dari solusi tersebut. 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. Hasil dari evaluasi pada tahap ini dimasukkan ke dalam populasi sementara [4].

3.3.5 Seleksi

Untuk mendapatkan solusi yang terbaik, program harus melakukan seleksi terhadap solusi dengan nilai *fitness* yang tergolong rendah [5]. Seleksi menggunakan metode *good fitness*, yaitu setengah dari jumlah populasi yang memiliki harga *fitness* terendah akan dihilangkan sehingga hanya selalu tersisa sekelompok solusi terbaik yang pernah diperoleh oleh program. Solusi yang tersisa dari hasil seleksi ini dikenal dengan nama populasi induk [6].

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 mutasi dari solusi induk. Tujuan pembangkitan solusi baru ini untuk menemukan alternatif solusi yang lebih baik dari solusi-solusi yang sudah diperoleh.

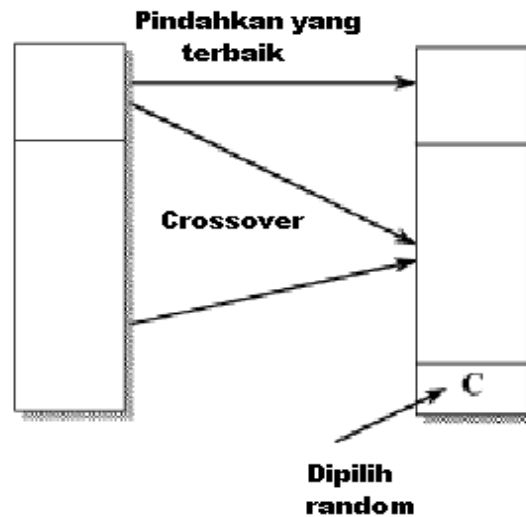
3.3.6 Reproduksi Kromosom Baru

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 sebelumnya yang telah dibahas. Dengan proses ini akan dihasilkan sekelompok populasi baru yang benar-benar berbeda dengan populasi induknya [6].

3.3.7 Mutasi

Proses ini dilanjutkan dari setengah populasi baru lainnya dimana akan dibangkitkan dengan cara mutasi, yaitu setengah dari populasi induk akan dipilih untuk diduplikasi. Pemilihan dapat dilakukan dengan metode *good fitness*, acak, maupun *roulette wheel*. 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 dengan

mempertahankan hal yang sudah sesuai dengan kriteria. Perubahan posisi beberapa mata kuliah ini (mutasi) dapat membuat solusi duplikasi memiliki nilai *fitness* yang lebih rendah maupun lebih tinggi daripada solusi induknya. Jika ternyata diperoleh solusi dengan nilai *fitness* yang lebih tinggi maka hal itulah yang diharapkan. Tetapi jika diperoleh solusi dengan nilai *fitness* yang lebih rendah maka pada iterasi berikutnya kemungkinan diperoleh solusi hasil mutasi yang lebih baik nilai *fitness*-nya daripada solusi induknya [6].



Gambar 2. Pengembangan mutasi

3.3.8 Kondisi Selesai

Terdapat tiga kondisi selesai yang dapat menghentikan proses algoritma pemrograman ini [7], yaitu:

- 1) Jika setelah beberapa generasi berturut-turut nilai *fitness* terbaik dari populasi tidak mengalami perubahan pada iterasi berikutnya.
- 2) Jika jumlah generasi atau iterasi maksimum telah tercapai.
- 3) Jika nilai *fitness* terbaik telah tercapai.
- 4) Jika salah satu kondisi di atas telah diperoleh maka iterasi akan dihentikan. Jika salah satu kondisi selesai ini belum tercapai maka program akan mengulang kembali proses ini atau iterasi dari langkah keempat, yaitu mutasi dan perhitungan *fitness* terhadap populasi baru tersebut.

4. KESIMPULAN

Secara teori bantuan Algoritma Genetik Biner dalam penyusunan penjadwalan kuliah dapat digunakan. Dengan diaplikasikan menggunakan salah satu bahasa pemrograman dalam ilmu optimasi, misalnya *Mathlab* atau *Labview*, program tersebut dapat mencari solusi penjadwalan pada waktu yang dapat digunakan, baik oleh dosen, kelas, maupun ruangan, yang terkait dalam suatu mata kuliah dengan waktu yang singkat. Dengan menggunakan metode *best fitness*, Algoritma Genetik akan selalu menunjukkan kenaikan *fitness* atau dengan kata lain generasi selanjutnya lebih baik atau minimal sama dengan generasi sebelumnya serta dapat menemukan solusi optimal dari suatu permasalahan.

REFERENSI

- [1]. Haupt, Randy L., “*Practical Genetic Algorithms*”, A John Wiley & Sons, Inc, 2004.
- [2]. Kusumadewi, Sri, “*Artificial Intelligence (Teknik dan Aplikasinya)*”, Graha Ilmu, Yogyakarta, 2003.
- [3]. Nugraha, Ivan, “*Aplikasi Algoritma Genetik untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*”, Institut Teknologi Bandung, Bandung, 2008.
- [4]. Santosa, Budi dan Paul Willy, “*Metoda Metaheuristik Konsep dan Implementasi*”, Guna Widya, 2011.
- [5]. Pohlheim, Hartmut, “*GEA Toolbox: Genetic and Evolutionary Algorithms: Principles, Methods, and Algorithms*”, Tersedia dari <http://www.geatbx.com/docu/algindex.html>.
- [6]. <http://www.obitko.com/tutorials/genetic-algorithms/>, diakses 4 Juni 2013.
- [7]. Erben, Wilhelm, “*A Hybrid Grouping Genetic Algorithm for Examination Timetabling*”, University of Applied Sciences, 2005.