

Optimasi Pembuatan Jadwal Pelajaran Menggunakan Algoritma Genetik Studi Kasus SMK Ma'arif Kota Mungkid

Gunawan Budi Sulisty
AMIK BSI Yogyakarta
gunawan.gnw@bsi.ac.id

Abstract - The activity of making a schedule on teaching and learning activities in a school is a complicated routine. This Scool Timetabling Problems (STP) is in addition to the teacher's view, which is likely to teach more than one lesson, considering the number of teachers who are still less than the number of lessons and classrooms. So there must be a solution so that a teacher does not have two different subjects on the same day and hour. One method that can be used to solve the problem is the genetic algorithm approach. Genetic algorithms are a computational approach to solving problems modeled by biological processes of evolution. It is expected that the use of genetic algorithm will be optimized scheduling that is the condition where there is the best combination for the subject pair and the teacher as a whole, there is no problem of scheduling clashes on the students side, as well as the availability of sufficient and suitable room facility for all subjects.

Keyword: Optimization, Schedule making, Teaching, Algorithm, Genetic

Abstract - Kegiatan pembuatan jadwal pada kegiatan belajar mengajar disuatu sekolah merupakan rutinitas yang rumit. Scool Timetabling Problems (STP) ini selain dilihat dari sisi guru, yang kemungkinan akan mengajar lebih dari satu pelajaran, mengingat jumlah guru yang masih kurang dibanding jumlah pelajaran dan ruang kelas. Sehingga harus dicari solusi agar seorang guru tidak mengampu dua mata pelajaran berbeda pada hari dan jam yang sama. Salah satu metode yang dapat digunakan untuk memecahkan masalah tersebut adalah pendekatan algoritma genetik. Algoritma genetik merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Diharapkan dengan digunakannya algoritma genetik akan diperoleh optimasi penjadwalan yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan mata pelajaran dan guru secara keseluruhan, tidak ada permasalahan bentrokan jadwal pada sisi siswa, serta ketersediaan ruang yang cukup dan sesuai secara fasilitas untuk seluruh mata pelajaran yang ada.

Keyword : Optimasi, pembuatan Jadwal, Mengajar, Algoritma, Genetik

1. PENDAHULUAN

Pembuatan penyusunan jadwal kegiatan belajar mengajar di sekolah merupakan pekerjaan yang tidak mudah. Terdapat berbagai aspek yang berkaitan dalam penjadwalan tersebut yang harus dilibatkan dalam pertimbangan di antaranya :

1. Setiap mata pelajaran hanya dapat diampu oleh guru yang berlatar belakang Pendidikan yang sama.
2. Jumlah jam harus mengikuti peraturan yang ditetapkan oleh Dinas Pendidikan.
3. Jumlah kelas dalam satu sekolahan harus diperhatikan, agar semua kelas terisi guru.
4. SMK Maarif saat memiliki 3 ruangan, Teknik Pemesinan, Teknik Kendaraan Ringan, Teknik Komputer dan Jaringan.
5. Beberapa pelajaran dapat diampu oleh guru yang dibidangnya, seperti agama, olahraga, Bahasa Indonesia, Bahasa Inggris, Matematika, Kimia dan pelajaran husus TKJ.

Di samping aspek-aspek di atas, dalam penyusunan jadwal pelajaran ini pun terdapat

sangat banyak kemungkinan yang selayaknya dicoba untuk menemukan pembuatan jadwal yang terbaik. Karena itu dibutuhkan metode optimasi yang dapat diterapkan untuk mengerjakan penyusunan mata pelajaran ini.

2. ALGORITMA GENETIK

2.1. Pengertian Algoritma Genetik

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih "alamiah". Salah satu aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi. Dalam tulisan ini akan dibahas teori dasar algoritma genetika beserta contoh aplikasinya dalam menyelesaikan suatu permasalahan optimasi kombinasi sederhana.

Teori Dasar Algoritma Genetika

Algoritma genetika yang dikembangkan oleh Goldberg adalah algoritma komputasi yang diinspirasi teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi aturan “yang kuat adalah yang menang”. Darwin juga menyatakan bahwa kelangsungan hidup suatu makhluk dapat dipertahankan melalui proses reproduksi, crossover, dan mutasi. Konsep dalam teori evolusi Darwin tersebut kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih “alamiah”.

Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai chromosome, sedangkan kumpulan chromosome-chromosome tersebut disebut sebagai populasi. Sebuah chromosome dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan.

Chromosome-chromosome tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi. Dalam tiap generasi chromosome-chromosome tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (fungsi_objektif) menggunakan ukuran yang disebut dengan fitness. Untuk memilih chromosome yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan seleksi. Proses seleksi chromosome menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu chromosome yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya.

Chromosome-chromosome baru yang disebut dengan offspring, dibentuk dengan cara melakukan perkawinan antar chromosome-chromosome dalam satu generasi yang disebut sebagai proses crossover. Jumlah chromosome dalam populasi yang mengalami crossover ditentukan oleh parameter yang disebut dengan crossover_rate. Mekanisme perubahan susunan unsur penyusun makhluk hidup akibat adanya faktor alam yang disebut dengan mutasi direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam chromosome dengan suatu nilai acak. Jumlah gen dalam populasi yang mengalami mutasi ditentukan oleh parameter yang dinamakan mutation_rate.

Setelah beberapa generasi akan dihasilkan chromosome-chromosome yang nilai gennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh algoritma genetika terhadap permasalahan yang ingin diselesaikan.

2.2. Aplikasi 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.

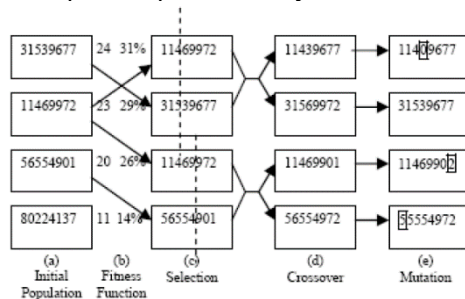
Mutasi

Pada mutasi, tiap lokasi menjadi sasaran mutase acak, dengan probabilitas independen yang kecil. Sebuah digit dimutasikan pada anak pertama, ketiga, dan keempat. Algoritma genetik mengkombinasikan suatu kecenderungan menaik dengan pengekplorasian 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 bertambahnya 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 kontinyu yang memberikan keuntungan yang konsisten. Algoritma genetik paling efektif dipakai bila schema-schema berkorespondensi menjadi komponen berarti dari sebuah solusi. Sebagai contoh, bila string adalah representasi dari sebuah antenna, maka schema merepresentasikan komponen-komponen dari antenna, seperti reflector dan deflector. Sebuah komponen yang baik cenderung akan berkerja baik pada rancangan yang berbeda. Ini enunjukkan bahwa penggunaan algoritma genetik yang benar memerlukan rekayasa yang baik pada representasinya.



Gambar 1. Penggunaan Algoritma Genetik

2.3. Algoritma Genetik

Algoritma genetika dikembangkan oleh John Holland pada tahun 1975. Algoritma genetika adalah suatu metode pencarian heuristik dan adaptive yang didasari prinsip evolusi biologi, algoritma ini dapat dilihat sebagai metode pencarian dan optimalisasi secara random dan berulang.

Beberapa keuntungan menggunakan

GA(Genetic Algorithm) :

- o Dapat melakukan optimalisasi baik pada variabel discrete atau continuous.
- o Bekerja secara simultan pada data yang besar.
- o Dapat menghasilkan sejumlah solusi yang optimal.
- o Melakukan proses encode dari variabel dan melakukan proses optimalisasi. Solusi dapat diperoleh dengan melakukan proses decode terhadap hasil optimalisasi.

Lima komponen dasar Algoritma Genetika :

- Representasi genetik dari solusi-solusi masalah
- Cara membentuk populasi awal dari solusi-solusi
- Fungsi evaluasi yang me-rate (rating) solusi-solusi berdasarkan fitness mereka

- Operator-operator genetik yang merubah komposisi genetik dari 'children' selama reproduksi
- Nilai-nilai untuk parameter algoritma genetika.

Jumlah perulangan(iterasi) yang dilakukan dalam proses GA.

Pseudocode Algoritma Genetika

Procedure Genetic Algorithms

begin

t <- 0;

inisialisasi P(t);

evaluasi P(t);

while (not syarat berhenti) do

begin

kombinasi-kembali P(t)

untuk membentuk C(t);

evaluasi C(t);

pilih P(t + 1) dari P(t) dan

C(t);

t <- *t* + 1;

end

End;

P : populasi

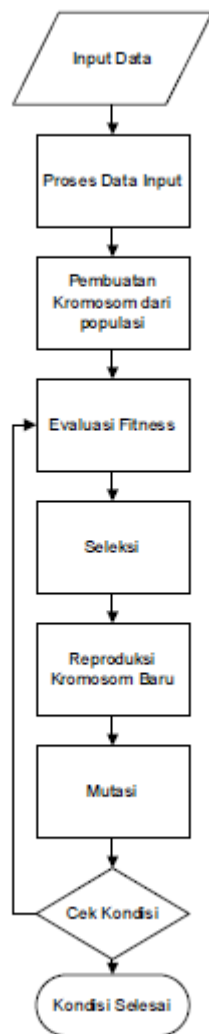
t : generasi / iterasi

C : offspring / anak

3. APLIKASI ALGORITMA GENETIK PADA OPTIMASI PEMBUATAN JADWAL

3.1. 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 Pelajaran
2. Tabel Guru
3. Tabel Kelas
4. Tabel Ruangan
5. Bobot *Fitness*
6. Kondisi Selesai

Tabel Mata Pelajaran berisikan daftar seluruh mata pelajaran yang akan dilaksanakan pada semester yang bersangkutan. Data yang perlu disertakan untuk setiap mata pelajaran yang ada adalah kelas peserta, jumlah jam, guru pengajar serta ruangan untuk mata pelajaran tersebut. Tabel Guru, Tabel Kelas, serta Tabel Ruangan adalah tabel waktu yang menginformasikan

waktu-waktu dari guru, kelas dan ruangan yang dapat digunakan untuk mata pelajaran yang bersangkutan.

Proses Data Input

Agar dapat diproses dalam algoritma ini Tabel Mata Pelajaran, Tabel Guru, Tabel Kelas dan Tabel Ruangan harus digabungkan terlebih dahulu menjadi Tabel Prioritas Mata Pelajaran. Untuk menjadwalkan suatu Mata Pelajaran, perlu mempertimbangkan jadwal waktu guru, kelas dan ruangan yang tersedia. Maka setiap mata pelajaran akan memiliki banyaknya pilihan jadwal yang berbeda. Dari proses ini diharapkan tidak ada mata pelajaran yang tidak dapat teralokasikan jadwalnya dikarenakan pada jadwal yang memungkinkan bagi mata pelajaran tersebut telah digunakan oleh mata pelajaran lainnya.

Pembuatan Kromosom dan Populasi

Berdasarkan urutan dari Tabel Prioritas Mata Pelajaran, setiap mata pelajaran akan dijadwalkan ke dalam Tabel Jadwal Mata Pelajaran secara acak. Agar diketahui apakah pada waktu tersebut guru, kelas, maupun ruangan dapat digunakan untuk melaksanakan pembelajaran, maka Tabel Guru, Tabel Ruang, dan Tabel Kelas untuk setiap mata pelajaran serta Tabel Mata Pelajaran harus dipetakan terlebih dahulu dalam Tabel Jadwal Mata Pelajaran. Sebelum menjadwalkan suatu mata pelajaran pada Tabel Jadwal Mata Pelajaran, algoritma akan mengecek terlebih dahulu kepada Tabel Jadwal Mata Pelajaran Bayangan untuk mengetahui apakah pada waktu tersebut dapat digunakan untuk pembelajaran 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 pelajaran; Terhadap mata pelajaran dengan bobot 3 jam, program dapat memecah mata pelajaran tersebut menjadi 2 atau 3 kelompok jam pelajaran jika waktu pembuatan jadwal yang ada tidak memungkinkan untuk dilaksanakannya mata pelajaran tersebut dalam satu waktu. Hal ini dibuat dengan tujuan memperluas kemungkinan alternatif pembuatan jadwal yang ada terutama pada mata pelajaran yang hanya memiliki sedikit alternatif pembuatan jadwal. Tetapi pemecahan mata pelajaran ini akan memperkecil nilai Fitness, sehingga kelak program akan cenderung

menyeleksi solusi penjadwalan yang memiliki pemecahan mata pelajaran 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 guru; Diinginkan agar tugas mengajar guru dapat terdistribusi merata di tiap hari kerjanya dengan tujuan agar performansi guru sewaktu mengajar dapat tetap dijaga optimal. Nilai *fitness* solusi akan berkurang jika dalam solusi tersebut terdapat guru yang memiliki tingkat mengajar terlalu tinggi dalam satu harinya.
4. Frekuensi belajar siswa; Seperti halnya pada guru, untuk menjaga performansi belajar siswa maka diharapkan tidak ada jadwal pelajaran yang terlalu padat dalam satu hari. Jika solusi menawarkan jadwal pelajaran kelas yang terlalu padat dalam satu hari, maka nilai *fitness* solusi yang berkurang. Kelas yang memiliki pelajaran lebih dari 5 jam pelajaran pada satu hari didefinisikan sebagai kelas yang memiliki frekuensi pelajaran yang tinggi.
5. Kedekatan antar mata pelajaran; Idealnya siswa memiliki waktu istirahat antar dua mata pelajaran yang ada dalam satu hari sehingga kelelahan siswa dalam mengikuti mata pelajaran pertama tidak mengganggu proses belajar pada mata pelajaran selanjutnya. Didefinisikan bahwa dua mata pelajaran yang berjarak kurang dari dua sks untuk satu kelasnya digolongkan sebagai mata pelajaran yang berdekatan dan dapat memperkecil nilai *fitness* dari solusi yang ditawarkan program. Walaupun ada waktu jeda antar dua mata pelajaran, tetapi diharapkan juga siswa tidak menunggu terlalu lama antar dua mata pelajaran tersebut. Rumus *fitness* yang digunakan adalah sebagai berikut:

$$Fitness = \frac{1}{B1xF1 + B2xF2 + B3xF3 + B4xF4 + B5xF5 + B6xF6} \quad (1)$$

de

- F1 = Banyaknya mata pelajaran yang dipecah
 F2 = Banyaknya waktu yang kosong
 F3 = Banyaknya frekuensi jam mengajar
 F4 = Banyaknya frekuensi jam pelajaran

F5 = Banyaknya mata pelajaran yang berdekatan

F6 = Banyaknya mata pelajaran yang berjauhan.

B1 = Bobot pemecahan mata pelajaran

B2 = Bobot waktu pagi yang kosong

B3 = Bobot frekuensi mengajar guru

B4 = Bobot frekuensi pelajaran kelas

B5 = Bobot mata pelajaran yang berdekatan

B6 = Bobot mata pelajaran yang berjauhan

4. HASIL PENGUJIAN

Algoritma pernah diuji dalam pembuatan jadwal mata pelajaran di jurusan TEKNIK KOMPUTER DAN JARINGAN semester I tahun ajaran 2014/2016 memakai LabView 2013 dimana melibatkan 37 mata pelajaran, 4 angkatan kelas, 8 ruang dan 15 guru. Algoritma ini diuji dalam 100 generasi, di mana setiap generasi terdiri dari 10 kromosom. Pada hasil akhir iterasi diperoleh solusi pembuatan jadwal dengan *fitness* 0,0083. Di mana tidak terdapat pemecahan mata pelajaran, tidak terdapat guru yang memiliki frekuensi mengajar tinggi dalam satu hari, tetapi terdapat 12 kelas kuliah per minggunya yang memiliki frekuensi mengajar tinggi. Dan terdapat 5 mata pelajaran yang berdekatan. Performansi akhir ini cukup baik dan menunjukkan bahwa algoritma genetik telah berhasil diaplikasikan untuk pembuatan jadwal.



Gambar 3. Tampilan Program Labview

5. KESIMPULAN

Dengan bantuan Algoritma Genetik pembuatan jadwal mata pelajaran dapat dioptimalkan. Program dapat mencari solusi pembuatan jadwal pada waktu yang dapat digunakan baik oleh guru, kelas maupun ruangan yang terlibat dalam suatu mata kuliah. Di samping itu, program dapat meminimalkan tingginya frekuensi mengajar seorang guru, frekuensi pelajaran suatu kelas dan faktor pengaruh lainnya. Proses pembuatan jadwal mata pelajaran menggunakan Algoritma Genetik ini dapat diterapkan pada kasus-kasus pembuatan jadwal dengan multi jurusan 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/2016-2017/cs/gafortimetabling.pdf> Tanggal Akses 20 Mei 2017
- [4] <http://ilmukomputer.com/2007/03/29/algoritma-genetikadan-contoh-aplikasinya/> . Tanggal Akses 20 Mei 2008
- [5] <http://lancet.mit.edu/~mbwall/presentations/IntroToGAs/> . Tanggal akses 20 Mei 2017
- [6] D. Dutta, R. Roy and K. Choudhury, 2013. Training Artificial Neural Network using Particle Swarm Optimization Algorithm. India: West Bengal: International Journal of Advanced Research in Computer Science and Software Engineering.
- [7] F. Gorunescu, Data Mining Concepts, Models and Techniques, Berlin: Springer, 2011.
- [8] Kusrini, luthfi taufiq Emha, (2009), Algoritma Data Mining, Penerbit Andi