OPTIMASI PENJADWALAN KEGIATAN BELAJAR MENGAJAR DENGAN ALGORITMA GENETIK

Usulan Skripsi S-1

Jurusan Matematika

Diajukan oleh

1. Novandry Widyastuti	M0105013
2. Astika Ratnawati	M0105025
3. Rahma Nur Cahyani	M0105059

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

2008

1. JUDUL

OPTIMASI PENJADWALAN KEGIATAN BELAJAR MENGAJAR DENGAN ALGORITMA GENETIK.

2. LATAR BELAKANG MASALAH

Penjadwalan kegiatan belajar mengajar dalam suatu kampus adalah hal yang rumit. Terdapat berbagai aspek yang berkaitan dalam penjadwalan tersebut yang harus dilibatkan antara lain terdapat jadwal-jadwal di mana dosen yang bersangkutan tidak bisa mengajar. Tidak boleh adanya jadwal kuliah yang beririsan dengan jadwal kuliah angkatan sebelumnya maupun sesudahnya, sehingga mahasiswa dapat mengambil mata kuliah angkatan sebelumnya maupun sesudahnya. Distribusi jadwal perkuliahan juga diharapkan dapat merata tiap harinya untuk setiap kelas. Pekerjaan penjadwalan mata kuliah ini akan semakin berat jika melibatkan semakin banyak kelas per angkatannya.

Permasalahan yang sering disebut dengan University Timetabling Problems (UTP) ini. selain dilihat dari sisi mahasiswa, juga harus dilihat dari sisi dosen, yaitu kemungkinan-kemungkinan dosen akan mengampu lebih dari satu mata kuliah yang ada, sebab ada kemungkinan jumlah mata kuliah dan jumlah dosen tidak sebanding, sehingga harus dipikirkan juga solusi agar dosen tidak mengampu dua mata kuliah berbeda pada hari dan jam yang sama. Selain itu, harus dipertimbangkan juga ketersediaan kelas sehingga kegiatan belajar dapat dilaksanakan. Di samping aspek-aspek di atas, dalam penyusunan jadwal kuliah ini pun terdapat sangat banyak kemungkinan yang selayaknya dicoba untuk menemukan penjadwalan yang terbaik. Karena itu dibutuhkan metode optimasi yang dapat diterapkan untuk mengerjakan penjadwalan mata kuliah 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. Diharapkan dengan digunakannya algoritma genetik akan diperoleh optimasi penjadwalan yaitu 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.

Berdasarkan uraian tersebut maka dalam skripsi ini akan dijelaskan bahwa 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.

3. PERUMUSAN MASALAH

Berdasarkan latar belakang masalah, disusun perumusan permasalahan yaitu bagaimana memperoleh optimasi penjadwalan dengan menggunakan algoritma genetik sehingga diperoleh 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.

4. BATASAN MASALAH

Dari analisis yang telah dilakukan dapat dirumuskan beberapa batasan masalah pada proses penjadwalan kuliah. Adapun batasan masalah tersebut adalah:

1. Spesifikasi permasalahan:
    a. Tidak boleh adanya bentrok waktu, ruang mengajar dosen dan perkuliahan mahasiswa.
    b. Adanya matakuliah dengan ruangan khusus (Praktikum).
    c. Durasi kuliah antara praktek dan teori yang berbeda per-sksnya.
    d. Adanya batas hari dalam satu minggu.
    e. Adanya batas jam kuliah dalam satu hari.
    f. Dosen dapat memilih jam mengajar.

g. Adanya waktu ruang tidak dapat digunakan.
h. Kapasitas ruangan sesuai jumlah mahasiswa.
i. Adanya waktu tertentu tidak boleh ada perkuliahan,

2. Batasan masalah secara teknis:
    a. Membangun aplikasi penjadwalan kuliah berbasis GUI (*Graphic User Interface*).
    b. Pemecahan permasalahan dengan metode algoritma genetika.
    c. Berbasis *Client Server* dan tetap memperhatikan sistem keamanan database.

5. TUJUAN PENELITIAN

Tujuan dan penelitian ini adalah memperoleh optimasi penjadwalan dengan menggunakan algoritma genetik sehingga diperoleh 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.

6. MANFAAT PENELITIAN

Adapun manfaat dari penelitian adalah untuk meningkatkan pemahaman tentang penggunaan algoritma genetik dalam memperoleh optimasi penjadwalan sehingga diperoleh 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.

7. METODOLOGI PENELITIAN

Langkah-langkah yang ditempuh dalam menyelesaikan penelitian sebagai berikut:

1. Studi literatur algoritma genetika untuk menyelesaikan permasalahan penjadwalan kuliah.

2. Mengidentifikasi permasalahan atau kendala yang dihadapi dalam menyusun jadwal kuliah.
3. Membuat rancangan database dan proses jadwal kuliah, meliputi:
    a. Membuat ERD (*Entity Relathionship Diagram*) untuk menggambarkan hubungan antara entitas.
    b. Membuat Skema Diagram.
    c. Membuat *Context Diagram*.
    d. Membuat DFD (*Data Flow Diagram*) Level 0 dan Level 1 untuk menggambarkan proses-proses yang terjadi dalam sistem penjadwalan kuliah.
4. Mengimplementasikan rancangan algoritma genetika pada modul program (analisis coding program).
5. Membangun aplikasi penjadwalan dengan database berdasarkan hasil analisis.
6. Mengevaluasi kinerja program.

8. TINJAUAN PUSTAKA

8.1 Pengertian Algoritma Genetika

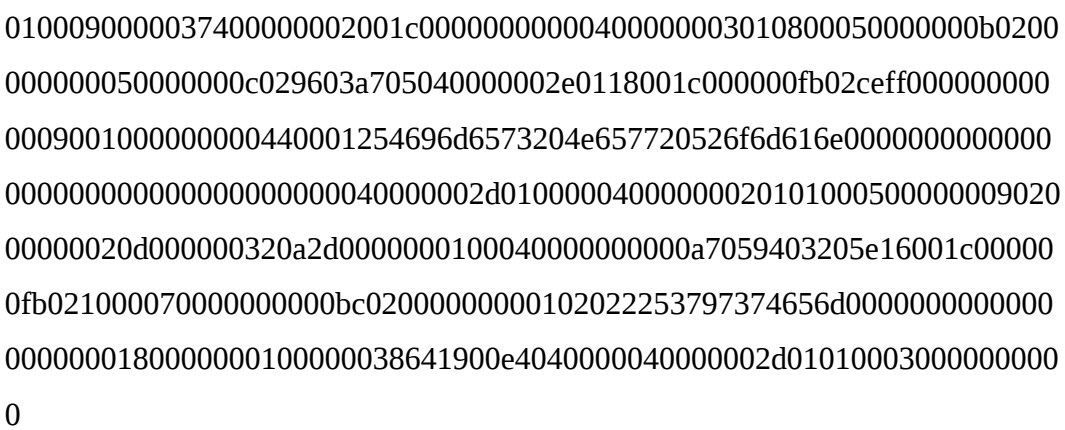
Algoritma genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma Genetika banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan lainnya. Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi. Solusi-solusi dari sebuah populasi diambil dan digunakan untuk membentuk populasi yang baru. Hal ini dimotivasi dengan harapan bahwa populasi yang baru dibentuk tersebut akan lebih baik daripada yang lama. Solusi-solusi yang dipilih untuk membentuk solusisolusi yang baru dipilih sesuai dengan *fitness* mereka masing-masing (Juniawati, 2003). Dalam buku dengan judul "*Adaption in Natural and Artificial System*" yang terbit pada tahun 1975, prinsip algoritma genetika diambil dari teori Darwin yaitu setiap makhluk hidup akan menurunkan satu atau beberapa karakter ke anak atau keturunannya (Bambrick, 1997). Di dalam proses tersebut dapat terjadi variasi yang disebabkan karena adanya mutasi, sehingga keturunan yang dihasilkan dapat

mempunyai kelebihan bahkan tidak memiliki kekurangan dari orangtuanya.

Setiap makhluk hidup akan mengalami seleksi alam, sehingga makhluk hidup yang mempunyai kemampuan untuk beradaptasi dengan lingkungan sekitarnya dapat bertahan sampai generasi selanjutnya. Semakin bagus atau sesuai *fitness* dari sebuah solusi maka solusi tersebut mempunyai peluang besar untuk dipilih. Proses ini dilakukan berulang sampai kondisi tertentu dipenuhi.

8.2 Dasar Algoritma Genetika

Algoritma genetika pertama kali dikemukakan oleh John Holland awal tahun 1975. John Holland mengemukakan bahwa algoritma genetika merupakan program komputer yang meniru proses evolusi alam. Kerangka dasar dari algoritma genetika sering disebut *Simple Genetic Algorithm* oleh John Holland dinyatakan sebagai berikut (Bambrick, 1997) (Gambar 2.1):



Gambar 2.1 *Simple Genetic Algorithm*

Deskripsi Algoritma Genetika

1. [*Start*] *Generate* populasi pertama secara random sebanyak n individu.
2. [*Fitness*] Evaluasi nilai *fitness* f(x) dari setiap individu x didalam populasi.
3. [New Populasi] Bentuk populasi baru dengan melakukan pengulangan langkah-langkah dibawah ini sehingga didapatkan populasi baru.
    a. [*Selection*] Pilih 2 individu sebagai orangtua dari

sebuah populasi sesuai dengan *fitness* mereka (Semakin baik *fitness*, maka semakin besar peluang mereka untuk dipilih).

b. [*Crossover*] Lakukan persilangan antara kedua orangtua sesuai dengan probabilitas crossover untuk membentuk keturunan yang baru. Jika tidak terjadi persilangan maka keturunan yang dihasilkan akan sama persis dengan orangtuanya.
c. [*Mutation*] Mutasi setiap keturunan yang baru sesuai dengan probabilitas mutasi di setiap gen.
d. [*Accepting*] Tempatkan keturunan yang baru sebagai populasi yang baru.

4. [*Replace*] Gunakan populasi yang baru dibentuk untuk menjalankan algoritma.
5. [*Test*] Jika kondisi akhir dipenuhi maka berhenti dan tampilkan solusi dari populasi.
6. [*Loop*] Kembali ke nomor 2.

8.3 Penerapan Algortima Genetika

8.3.1 Membangun Generasi Awal

Langkah pertama dalam algoritma ini adalah membentuk sejumlah populasi awal yang digunakan untuk mencari penyelesaian optimal. Populasi awal yang dibangun dalam tugas akhir ini dengan menggunakan bilangan random (acak) dengan range bilangan yang telah ditentukan (Mitchell 2007).

8.3.2 Representasi Kromosom

Algoritma genetika tidak beroperasi dengan penyelesaian asli dari suatu masalah tetapi beroperasi dengan dengan penyelesaian yang telah di representasikan. Representasi kromosom merupakan proses pengkodean dari penyelesaian asli dari suatu permasalahan. Pengkodean kandidat penyelesaian ini disebut dengan kromosom. Pengkodean tersebut meliputi penyandian gen, dengan

satu gen mewakili satu variabel (Mitchell 2007).

8.3.3 Fungsi *Fitness*

Fungsi *fitness* digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan. Fungsi *fitness* tersebut sebagai berikut (Hermanto 2003).

$$Fitness = \frac{1}{1 + penalty}$$

Dari persamaan diatas nilai *fitness* ditentukan oleh nilai *penalty*. *Penalty* tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai *penalty* berbanding terbalik dengan nilai *fitness*, semakin kecil nilai *penalty* (jumlah pelanggaran) semakin besar nilai *fitness*nya. Jadi fungsi *fitness* :

$$\frac{1}{1 + \sum Bp + \sum Np}$$

Keterangan:

Bp=Bobot pelanggaran

Np=Indikator Pelanggaran

8.3.4 Seleksi

Setiap kromosom yang terdapat dalam populasi akan melalui proses seleksi untuk dipilih menjadi orangtua. Sesuai dengan teori Evolusi Darwin maka kromosom yang baik akan bertahan dan menghasilkan keturunan yang baru untuk generasi selanjutnya.

Ada beberapa metode seleksi (Juniawati, 2003), yaitu

1. *Steady-State Selection*

Pemikiran utama dari metode seleksi ini adalah sebagian kromosom dari generasi lama tetap bertahan atau berada di generasi selanjutnya. Algoritma genetika menerapkan pemikiran tersebut dengan cara, didalam setiap generasi sejumlah kromosom yang mempunyai nilai *fitness* tinggi dipilih untuk diproses untuk menghasilkan keturunan yang baru sedangkan kromosom dengan nilai *fitness* rendah dibuang.

2. Elitism

Pembentukan populasi baru dengan *crossover* dan mutasi ada kemungkinan kromosom yang paling baik hilang. Oleh karena itu metode ini sebagai tahap awal memasukkan kromosom dengan nilai *fitness* yang paling baik atau beberapa kromosom dengan nilai fitnes yang tinggi atau cukup tinggi dari generasi yang lama kedalam generasi yang baru. Kemudian sisa kromosom dalam generasi yang baru diperoleh dengan cara reproduksi biasa.

3. Roulette Wheel Selection

Kromosom dipilih berdasarkan nilai *fitness*, semakin besar nilai *fitness* maka kromosom tersebut mempunyai peluang untuk dipilih beberapa kali.

8.3.5 Operator Genetika

Operator genetika dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Ada dua operator genetika yaitu *crossover* dan *mutation* (Juniawati, 2003).

1. Persilangan (*Crossover*)

Operator persilangan merupakan operasi yang bekerja untuk menggabungan dua kromosom orangtua (*parent*) menjadi kromosom baru (*offspring*). Tidak semua kromosom mengalami persilangan. Jumlah kromosom dalam populasi yang mengalami persilangan ditentukan oleh paramater yang disebut dengan *crossover rate* (probabilitas persilangan) .

Jenis operator persilangan yaitu:

a. One point crossover

Sebuah titik *crossover* dipilih, selanjutnya string biner mulai dari awal kromosom sampai dengan titik tersebut disalin dari salah satu

orangtua ke keturunannya, kemudian sisa bit keturunan disalin dari orangtua yang kedua.

Contoh : **<u>11001</u>**011 + 11011**<u>111</u>** = **11001111.**

b. Two point crossover

Dua titik *crossover* dipilih, selanjutnya string biner mulai dari awal kromosom sampai dengan titik *crossover* pertama disalin dari salah satu orangtua ke keturunannya kemudian mulai dari titik *crossover* pertama sampai dengan titik kedua disalin dari orangtua kedua. Sisanya disalin dari orangtua pertama.

Contoh : **<u>11</u>**0010**<u>11</u>** + 11**<u>011</u>**11 = **11011111**

2. Mutasi

Setelah *crossover* dilakukan, proses reproduksi dilanjutkan dengan mutasi. Hal ini dilakukan untuk menghindari solusi-solusi dalam populasi mempunyai nilai lokal optimum. Mutasi adalah proses mengubah gen dari keturunan secara random. Untuk pengkodean biner maka mutasi mengubah bit 0 menjadi bit 1 dan bit 1 menjadi bit 0.

Contoh : 1**<u>1</u>**001001 ? 1**<u>0</u>**001001

Tidak setiap gen selalu dimutasi tetapi mutasi dikontrol dengan probabilitas tertentu yang disebut dengan *mutation rate* (probabilitas mutasi) dengan notasi Pm. Jenis operator mutasi antara lain:

a. Mutasi Terarah

Mutasi terarah tergantung dari informasi gen. Informasi gen tersebut berupa nilai pelanggaran gen (*violation score*). Ini berarti bahwa setiap gen mempunyai peluang yang berbeda untuk terjadi mutasi. Gen yang mempunyai nilai pelanggaran yang lebih besar maka gen tersebut mempunyai peluang untuk terjadi mutasi. Mutasi ini menghubungkan nilai pelanggaran relatif (nilai pelanggaran suatu gen dibagi dengan nilai pelanggaran total suatu kromosom) dengan probabilitas terjadinya mutasi dari suatu gen pada kromosom. Hubungan tersebut dinyatakan secara matematis sebagai berikut:

$$nr(i) = \frac{n(i)}{1 + n_{total}}$$

$$pm(i) = (1 + nr(i))^2 \, pm$$

Keterangan persamaan:

nr(i) : nilai pelanggaran relative gen ke-i.

ntotal : nilai pelanggaran total kromosom.

pm(i) : probabilitas mutasi gen ke-i.

pm : probabilitas mutasi

b. Mutasi Biasa

Mutasi ini tidak tergantung dari informasi gen. Setiap gen mempunyai peluang yang sama untuk terjadi mutasi.

8.3.6 Parameter Genetika

Pengoperasian algoritma genetika dibutuhkan 4 parameter (Juniawati, 2003) yaitu:

1. Probabilitas Persilangan (*Probability Crossover*)

   Menunjukkan kemungkinan *crossover* terjadi antara 2 kromosom. Jika tidak terjadi *crossover* maka keturunannya akan sama persis dengan kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika probabilitas *crossover* 100% maka semua keturunannya dihasilkan dari *crossover*. *Crossover* dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik.

2. Probabilitas Mutasi (*Probability Mutation*)

   Menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah kromosom. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah *crossover* tidak berubah. Jika terjadi mutasi bagian kromosom akan berubah. Jika probabilitasnya 100 %, semua kromosom dimutasi. Jika probabilitasnya 0%, tidak ada yang mengalami mutasi.

3. Jumlah Individu

Menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetika akan mempunyai sedikit variasi kemungkinan untuk melakukan *crossover* antara orangtua karena hanya sebagian kecil dari *search space* yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetika akan berjalan lambat.

4. Jumlah Populasi

Menentukan jumlah populasi atau banyaknya generasi yang dihasilkan, digunakan sebagai batas akhir proses seleksi, persilangan, dan mutasi.