

**PENJADWALAN UJIAN AKHIR SEMESTER
DENGAN ALGORITMA GENETIKA
(STUDI KASUS JURUSAN TEKNIK INFORMATIKA UNESA)**

Anita Qoiriah

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya, anitaqoi@yahoo.com

Abstrak

Penyusunan jadwal merupakan suatu permasalahan yang sering terjadi dalam kehidupan kampus. Terdapat banyak kegiatan kampus yang membutuhkan penjadwalan karena adanya pengaruh keterbatasan ruang, kegiatan dosen, kegiatan mahasiswa dan sebagainya. Ujian akhir semester dilaksanakan di jurusan Teknik Informatika Unesa dalam setiap akhir semester untuk mengevaluasi proses belajar dalam setiap semesternya. Waktu pelaksanaan ujian akhir semester sudah ditentukan sesuai dengan jadwal akademik universitas yang pelaksanaannya sekitar 2 minggu atau 10 hari.

Algoritma Genetika adalah program komputer yang mensimulasikan proses evolusi, dengan menghasilkan kromosom-kromosom dari tiap populasi secara acak dan memungkinkan kromosom tersebut berkembang biak sesuai dengan hukum-hukum evolusi yang nantinya diharapkan akan dapat menghasilkan kromosom prima atau yang lebih baik. Kromosom ini merepresentasikan solusi dari permasalahan yang diangkat, sehingga apabila kromosom yang baik tersebut dihasilkan, maka diharapkan solusi yang baik dari permasalahan tersebut juga didapatkan. Algoritma Genetika bisa digunakan untuk menyelesaikan permasalahan yang tidak mempunyai metode penyelesaian dengan rumusan yang tepat, ataupun jika ada rumusannya, masih diperlukan waktu yang lama untuk menyelesaikannya, biasanya permasalahan tersebut sangat beragam dan Kompleks.

Untuk menerapkan Algoritma Genetika kedalam penjadwalan ujian akhir semester diperlukan beberapa proses sehingga jadwal akan terbentuk dengan optimal sesuai dengan aturan-aturan yang diberikan. Aturan pada pembuatan penjadwalan ujian semester ini matakuliah untuk kelas yang sama tidak diperbolehkan dilaksanakan bersamaan, satu ruang hanya untuk satu matakuliah, untuk matakuliah MKU (Matakuliah Umum) jadwal sudah ditetapkan dan tidak boleh dirubah., matakuliah praktikum berada di laboratorium, matakuliah yang sama dengan dosen yang sama untuk kelas yang berbeda boleh dilaksanakan bersama karena pengawas ujian tidak harus dosen yang bersangkutan.

Langkah-langkah Algoritma Genetika untuk menyelesaikan masalah penjadwalan adalah inialisasi populasi pertama dilakukan secara acak, penghitungan nilai fitness, proses seleksi dengan menggunakan *Roulette Wheel Selection*, proses crossover dengan probabilitas 0,6. *Crossover* yang dipilih adalah *Single-Point Crossover*, mutasi dimana probabilitas mutasi dipilih 0,001. Mutasi hanya diperbolehkan untuk gen-gen yang bermasalah dan terdapat penurunan nilai fitness.

Kata Kunci : Algoritma Genetika, ujian akhir semester, aturan-aturan, seleksi, *fitness*, *crossover*, mutasi.

PENDAHULUAN

Penyusunan jadwal merupakan suatu permasalahan yang sering terjadi dalam kehidupan kampus. Terdapat banyak kegiatan kampus yang membutuhkan penjadwalan karena adanya pengaruh keterbatasan ruang, kegiatan dosen, kegiatan mahasiswa dan sebagainya. Beberapa jenis penjadwalan di kampus antara lain penjadwalan kuliah yang harus dibuat dalam setiap semester, jadwal ujian sub sumatif, jadwal ujian akhir dan sebagainya.

Ujian akhir semester dilaksanakan di jurusan Teknik Informatika Unesa dalam setiap akhir semester untuk mengevaluasi proses belajar dalam setiap semesternya. Waktu pelaksanaan ujian akhir semester sudah ditentukan sesuai dengan jadwal akademik universitas yang pelaksanaannya sekitar 2 minggu atau 10 hari.

Selama ini pembuatan jadwal ujian akhir semester dilakukan secara manual sehingga cukup merepotkan. Karena dalam waktu 10 hari semua matakuliah harus diujikan dengan kondisi ruang yang terbatas. Selain itu waktu pelaksanaan ujian dalam satu hari juga biasanya dibatasi hanya tiga jam ujian.

Algoritma genetika merupakan salah satu algoritma optimasi yang termasuk dalam algoritma probalistik. Dimana dalam algoritma probalistik berusaha menemukan solusi yang bagus tanpa melebihi batasan waktu yang disediakan. Solusi yang bagus belum tentu yang paling optimum(global optimum) tetapi sudah dapat diterima oleh user(Suyanto, 2010).

Ruang masalah penjadwalan cukup besar sehingga penyelesaiannya lebih sesuai menggunakan algoritma probalistik. Algoritma Genetika merupakan algoritma probalistik yang paling mudah diimplementasikan untuk menyelesaikan masalah penjadwalan karena representasi individunya lebih fleksibel untuk berbagai macam masalah dan bisa dimodifikasi sesuai masalah yang dihadapi. Masalah penjadwalan dalam perkuliahan termasuk ujian akhir semester membutuhkan pengkodean representasi individu yang agak rumit karena berukuran besar. Sehingga pengkodean yang kurang tepat bisa mempengaruhi proses evolusi yang terjadi, terutama rekombinasi dan mutasi(Suyanto, 2010).

Dalam penelitian ini akan dibuat penjadwalan ujian akhir semester dengan menerapkan Algoritma Genetika

agar didapat jadwal yang sesuai dengan aturan-aturan yang ditentukan pada jurusan Teknik Informatika Unesa.

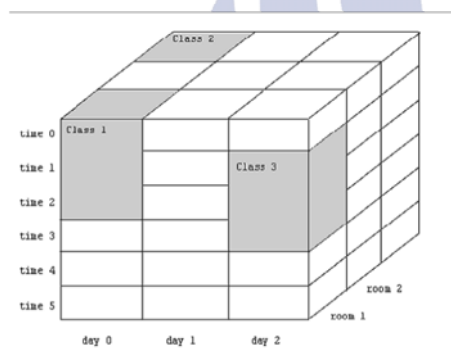
KAJIAN PUSTAKA

Penjadwalan

Menurut Leon Bambrick penjadwalan sesungguhnya adalah penyusunan jadwal yang menyesuaikan dengan sejumlah batasan/*constraint*. *Constraint* yang digunakan sangat tergantung dengan permasalahan dan kesepakatan orang-orang yang terlibat didalam permasalahan tersebut.

Terdapat 2 macam kategori *constraint*, *hard constraint* dan *soft constraint*. *Hard constraint* adalah *constraint* dimana penjadwalan tidak akan bekerja jika *constraint* tersebut dilanggar, contoh dosen mengajar dua matakuliah dalam waktu yang sama. *Soft constraint* merupakan *constraint* yang boleh dilanggar tetapi harus seminimal mungkin. Contoh, ukuran ruang kelas harus sesuai dengan jumlah peserta kuliah. (leon Bambrick, 1997).

Penjadwalan kuliah dapat digambarkan dalam bentuk 3 dimensi seperti gambar 1(Asif Ansari, 2014).



Gambar1. Penjadwalan yang dipresentasikan dalam bentuk 3 dimensi

Algoritma Genetika

Pada dasarnya Algoritma Genetika adalah program komputer yang mensimulasikan proses evolusi, dengan menghasilkan kromosom-kromosom dari tiap populasi secara acak dan memungkinkan kromosom tersebut berkembang biak sesuai dengan hukum-hukum evolusi yang nantinya diharapkan akan dapat menghasilkan kromosom prima atau yang lebih baik. Kromosom ini merepresentasikan solusi dari permasalahan yang diangkat, sehingga apabila kromosom yang baik tersebut dihasilkan, maka diharapkan solusi yang baik dari permasalahan tersebut juga didapatkan. Algoritma Genetika bisa digunakan untuk menyelesaikan permasalahan yang tidak mempunyai metode penyelesaian dengan rumusan yang tepat, ataupun jika ada rumusnya, masih diperlukan waktu yang lama untuk menyelesaikannya, biasanya permasalahan tersebut sangat beragam dan Kompleks(Suyanto, 2005).

Ada 6 komponen utama dalam Algoritma Genetika(Sri kusumadewi, 2003) :

1. Teknik penyandian

Teknik penyandian meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom.

Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk: string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

2. Prosedur inialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inialisasi dilakukan secara acak dengan memperhatikan domain solusi dan *constraint* yang digunakan.

3. Fungsi evaluasi

Terdapat 2 hal dalam evaluasi kromosom yaitu : evaluasi fungsi obyektif dan konversi fungsi obyektif kedalam fungsi *fitness*.

4. Seleksi

Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Terdapat beberapa metode dalam seleksi induk yaitu : *Rang-based fitness assignment*, *Roulette wheel sampling*, *Stochastic universal sampling*, *Local selection*, *Truncation selection*, *Tournament selection*.

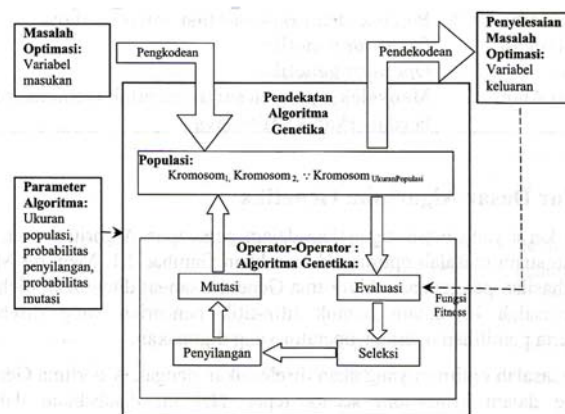
5. Operator genetika

Terdapat 2 operator genetika yaitu: operator untuk melakukan rekombinasi dan operator mutasi.

6. Penentuan parameter

Yang dimaksud dengan parameter disini adalah parameter kontrol Algoritma Genetika, yaitu ukuran populasi, peluang *crossover* dan peluang mutasi. Nilai parameter ini ditentukan berdasar permasalahan yang akan dipecahkan.

Kerangka kerja yang biasa digunakan dalam penerapan Algoritma Genetika untuk menyelesaikan suatu masalah optimasi ditunjukkan gambar 2 (Zainudin Zuhri, 2014). Dari proses Algoritma Genetika ini akan menghasilkan satu individu terbaik yang bertahan dalam proses regenerasi.



Gambar 2. Kerangka kerja Algoritma Genetika

Langkah awal dalam Algoritma Genetika ini adalah dengan melakukan pengkodean kromosom sesuai dengan permasalahan yang akan dipecahkan.

Kemudian dilanjutkan dengan pembentukan populasi awal yang dibangkitkan secara acak sesuai dengan representasi masalah. Ukuran populasi ditentukan sesuai dengan permasalahan. Untuk permasalahan yang memiliki kawasan solusi yang cukup besar, De Jong merekomendasikan sebesar 50 (Sri Kusumadewi, 2003).

Selanjutnya operator-operator genetika akan menggabungkan informasi genetis dari unsur-unsur populasi untuk membentuk populasi generasi berikutnya. Setiap kromosom mempunyai nilai *fitness* yang setara dengan nilai penyelesaian masalah. Pada generasi berikutnya, nilai *fitness* kromosom sebagai representasi dari penyelesaian masalah, diharapkan bertambah semakin bagus. Dari nilai *fitness* setiap kromosom akan dilakukan seleksi untuk memilih induk yang selanjutnya akan dilakukan proses penyilangan (*crossover*) dilanjutkan dengan mutasi. Untuk permasalahan yang memiliki kawasan solusi yang cukup besar, De Jong merekomendasikan nilai probabilitas *crossover* 0,6 dan mutasi sebesar 0,001 (Sri Kusumadewi, 2003).

METODE REKAYASA

Untuk menerapkan Algoritma Genetika kedalam penjadwalan akhir semester diperlukan beberapa proses sehingga jadwal akan terbentuk dengan optimal sesuai dengan aturan-aturan yang diberikan.

Aturan-aturan Penjadwalan Ujian

Aturan pada pembuatan penjadwalan ujian semester ini :

1. Matakuliah untuk Kelas yang sama tidak diperbolehkan dilaksanakan bersamaan.
2. Satu ruang hanya untuk satu matakuliah
3. Untuk Matakuliah MKU (Matakuliah Umum) jadwal sudah ditetapkan dan tidak boleh diubah.
4. Matakuliah praktikum berada di laboratorium
5. Matakuliah yang sama dengan dosen yang sama untuk kelas yang berbeda boleh dilaksanakan bersama karena pengawas ujian tidak harus dosen yang bersangkutan

Representasi Individu

Untuk merepresentasikan individu dalam kasus ini dilakukan dengan membuat tabel data matakuliah yang diujikan beserta kelasnya. Selanjutnya setiap data tersebut akan mempunyai nomer yang akan menjadi nomer dari gen yang membentuk kromosom. Gen-gen tersebut akan diisi dengan slot jadwal

Algoritma Genetika dalam Penjadwalan Ujian

Penerapan algoritma dalam pembuatan jadwal ujian akhir semester ini sesuai dengan alur pada Algoritma Genetika seperti pada gambar 3:

1. Membangkitkan populasi Awal

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak sehingga membentuk suatu populasi. Jumlah populasi yang disarankan 50.

2. Menghitung nilai *fitness*

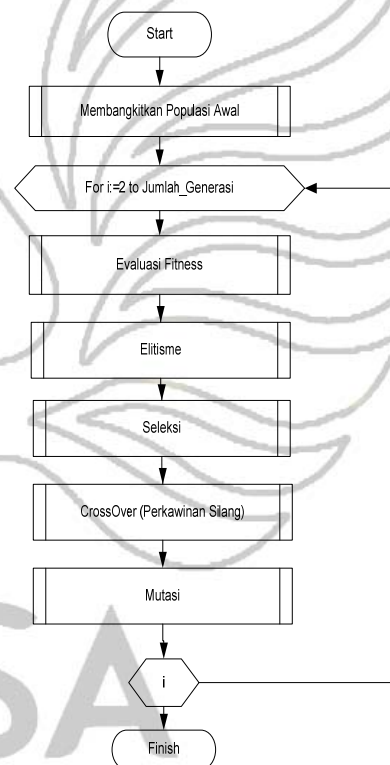
Dalam permasalahan ini fungsi obyektif merupakan nilai *fitness* yang didapat dari bobot setiap aturan yang ditentukan. Nilai *fitness* ini yang akan menyatakan baik atau tidaknya suatu solusi. *Fitness* yang semakin besar merupakan solusi yang paling baik karena nilai *fitness* diambil dari kebalikan nilai solusi ditambah bilangan yang mendekati nol (<0).

3. Memilih populasi terbaik

Proses seleksi individu menggunakan metode roulette wheel. Seleksi dilakukan untuk mendapatkan calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik juga. Seleksi menggunakan metode *roulette wheel* (roda roulette) akan memilih nilai dari *fitness*. Semakin tinggi nilai *fitness* maka semakin besar kemungkinan untuk terpilih.

4. Penyilangan/*Crossover*

Crossover merupakan salah satu komponen dalam Algoritma Genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Proses *crossover* dilakukan pada setiap individu dengan probabilitas yang disarankan 0,6.



Gambar 3. Flowchart Algoritma Genetika Penjadwalan Ujian Akhir Semester

5. Mutasi

Proses mutasi dilakukan terhadap individu terpilih. Mutasi merupakan operator yang menukar nilai gen dengan nilai kebalikannya dengan suatu probabilitas tertentu. Probabilitas mutasi diset 0,001.

PEMBAHASAN

Di dalam aplikasi penjadwalan ujian akhir semester ini beberapa data yang penting adalah data mata kuliah, data jam kuliah, data ruang kelas, aturan-aturan yang akan diterapkan. Dari data-data tersebut maka dapat dilakukan proses pembuatan jadwal sesuai dengan langkah-langkah dalam Algoritma Genetika.

a. Mata Kuliah

Data-data mata kuliah mulai dari kode mata kuliah, nama mata kuliah, SKS, Semester angkatan dan prodi. Jumlah data matakuliah rata-rata dalam satu semester di program studi PTI sejumlah 31 matakuliah, di program studi D3 Manajemen Informatika sejumlah 33 matakuliah.

b. Waktu Ujian

Ujian biasanya dilaksanakan selama 10 hari dengan 3 kali ujian dalam satu hari. Dimana 1 jam ujian dilaksanakan selama 100 menit. Jam 1 dimulai dari jam 07.00-08.40, jam ke 2 pukul 09.00-10.40 dan jam ke 3 pukul 11.00-12.40.

c. Ruang

Ruang di jurusan Teknik Informatika yang dapat digunakan untuk ujian sebanyak 5 kelas dan 4 laboratorium.

d. Kelas

Jumlah kelas setiap angkatan di jurusan Teknik Informatika rata-rata 3 kelas untuk S1 PTI dan 3 kelas untuk D3 MI, dimana kondisi sekarang D3 MI terdapat 3 angkatan dan S1 PTI mempunyai 3 angkatan.

e. Representasi Individu

Dalam penjadwalan ujian akhir semester ini kromosom mempresentasikan sebuah jadwal ujian akhir semester yang utuh untuk semua matakuliah dan semua kelas perkuliahan.

Data matakuliah akan dimasukkan kedalam gen tersebut merupakan urutan dari pertemuan kuliah sejumlah 94 untuk D3 MI dan 62 untuk S1 PTI. Sehingga total pertemuan kuliah sejumlah 156 kelas matakuliah.

Untuk merepresentasikan kromosom dalam penjadwalan ini, gen yang membentuk kromosom berisi slot waktu dan ruang. Sedang gen sendiri merupakan urutan kelas perkuliahan.

Basis data matakuliah beserta kelas seperti pada tabel 1. Urutan dari basis data tersebut akan menjadi urutan gen yang membentuk sebuah kromosom. Sehingga sebuah kromosom untuk proses penjadwalan kuliah pada semester gasal 2014/2015 di jurusan Teknik Informatika terdiri dari 156 gen.

Sedangkan nilai yang terdapat pada sebuah gen merupakan slot ruang dan waktu. Slot ruang dan waktu sejumlah $9(\text{kelas}) \times 3(\text{jam ujian}) \times 10(\text{hari ujian}) = 270$. Slot dari ruang dan waktu seperti pada tabel 2. Sehingga nilai angka 1 sampai 270 akan diisikan kedalam gen yang menunjukkan slot ruang dan waktu untuk matakuliah

yang dikodekan sebagai nomer gen Bentuk gen seperti gambar 2.

Dengan bentuk representasi seperti ini maka proses mutasi dan *crossover* lebih mudah untuk dilakukan.

Tabel 1. Basisdata Pertemuan Kuliah

No.	Kode	Nama Mata Kuliah	Kls	SKS	Juml Mhs
1.	97324213	Aljabar Linear dan Matriks	PTI13A	2	46
2.	97324213	Aljabar Linear dan Matriks	PTI13B	2	50
3.	97324215	Analisis Perancangan Sistem	PTI12A	2	33
4.	97324215	Analisis Perancangan Sistem	PTI12B	2	32
5.	90320202	Bahasa Inggris I	PTI14A	2	28
6.	90320202	Bahasa Inggris I	PTI14B	2	30
....					
156.	97324205	Dasar Dasar pemrograman	PTI14A	2	29

Tabel 2. Slot Ruang dan Waktu

Ruang 1

	Senin	Selasa	Rabu	Kamis	Jumat
1	Slot 1	Slot 4			
2	Slot 2	Slot 5			
3	Slot 3	...			

Ruang 9

	Senin	Selasa	Rabu	Kamis	Jumat
1					Slot 268
2				Slot 266	Slot 269
3				Slot 267	Slot 270

80	122	22	10		2
Gen 1	Gen 2	Gen 3	Gen 4		Gen 156

Gambar 4. Contoh Bentuk Gen

f. Fungsi Fitness

Fungsi *fitness* dalam penjadwalan ini berdasarkan *constraint* yang digunakan dalam pembentukan jadwal ujian akhir semester. Pada setiap pelanggaran *constraint* maka akan terdapat nilai penalti seperti pada tabel 3

Tabel 3. Penalti *Constraint*

No	Batasan/ <i>Constraint</i>	Penalti
1	Bentrok kelas	1000
2	Tipe ruang tidak sesuai	500
3	Ruang bentrok	1000
4	Sebuah kelas ujian melebihi 2 kali dalam sehari	50

Rumus *fitness* seperti berikut:

$$f = \frac{1}{(B + a)}$$

Dimana:

$$B = \sum_{i=1}^N JiPi$$

Ji merupakan jumlah pelanggaran batasan ke i . Sedang untuk Pi adalah nilai penalti untuk pelanggaran yang ke i . N adalah banyaknya batasan. a adalah suatu bilangan yang dianggap sangat kecil untuk menghindari pembagian dengan 0.

g. Proses penjadwalan

Setelah kromosom dan fungsi *fitness* didapatkan selanjutnya melakukan proses Algoritma Genetika seperti langkah-langkah yang sudah dijelaskan pada metode rekayasa seperti gambar ...

Tahap proses penjadwalan :

1. Menempatkan matakuliah MKU sesuai yang telah ditentukan. Gen yang sudah ditempati oleh matakuliah MKU diberi tanda sehi ngga tidak dapat diberi matakuliah lain. Gen-gen untuk matakuliah MKU tidak ikut dalam proses Algoritma Genetika.
2. Menempatkan mata kuliah praktikum secara acak pada laboratorium dengan memperhatikan *constraint* yang berhubungan dengan mahasiswa. Untuk memperbaiki pelanggaran *constraint* hanya dilakukan dengan mutasi .
3. Melakukan proses penjadwalan dengan algoritma genetika untuk matakuliah teori:
 - a. Inisialisasi populasi pertama dilakukan secara acak, banyaknya individu dalam populasi sebaiknya dibuat tidak terlalu banyak karena kromosom yang cukup panjang menggunakan banyak memori, berdasar kajian pustaka disarankan populasi sebesar 50.
 - b. Selanjutnya masing-masing individu dihitung nilai *fitness*nya
 - c. Kemudian dilanjutkan proses seleksi dengan menggunakan *Roulette Wheel Selection*. Dengan proses seleksi ini maka nilai *fitness* yang tinggi kemungkinan besar akan terpilih.
 - d. Induk yang didapat dari hasil seleksi kemudian dilakukan proses crossover dengan probabilitas 0,6. *Crossover* yang dipilih adalah single-point crossover. Proses crossover seperti potongan program pada gambar 5.

```
bool first = rand() % 2 == 0;
for( int i = 0; i < size; i++ )
{
    if( first )
    {
        n->kelas.insert(pair<jadw*,int>((*cr1).first,(*cr1).second));
        for( int i=(*cr1).first->GetDuration() - 1; i >= 0; i-- )
            n->slot[(*cr1).second + i ].push_back((*cr1).first);
    }
    else
    {
        n->kelas.insert(pair<jadw*,int>((*cr2).first,(*cr2).second));
        for( int i = (*cr2).first->GetDuration() - 1; i >= 0; i-- )
            n->slot[(*cr2).second + i ].push_back( (*cr2).first );
    }
    if( cp[ i ] )
    {
        first = !first;
        cr1++;
        cr2++;
    }
}
```

Gambar 5. Kode program proses *crossover*

- e. Dilanjutkan dengan mutasi dimana probabilitas mutasi dipilih 0,001. Mutasi hanya diperbolehkan untuk gen-gen yang bermasalah dan terdapat penurunan nilai *fitness*. Proses mutasi dapat dilihat sepertipotongan program pada gambar 6

```
for( int i = ukuran; i > 0; i-- )
{
    int mpos = rand() % jmlKelas;
    int pos1 = 0;
    hash_map<jadw*,int>::iterator it = kelas.begin();
    for( ; mpos > 0; it++, mpos-- ) |;
    pos1 = (*it).second;
    jadw* jdl = (*it).first;

    for( int i = dur - 1; i >= 0; i-- )
    {
        list<jadw*> cl = slot[ pos1 + i ];
        for( list<jadw*>::iterator it = cl.begin(); it != cl.end(); it++ )
        {
            if( *it == jdl )
            {
                cl.erase( it );
                break;
            }
        }
        slot.at( pos2 + i ).push_back( jdl );
    }
    kelas[ ccl ] = pos2;
}
```

Gambar 6. Kode program proses mutasi

KESIMPULAN DAN SARAN

Kesimpulan

Algoritma genetika sering digunakan untuk menyelesaikan permasalahan yang tidak mempunyai metode penyelesaian dengan rumusan yang tepat, ataupun jika ada rumusannya masih diperlukan waktu yang lama untuk menyelesaikannya, biasanya permasalahan tersebut sangat beragam dan kompleks. Dalam pembuatan jadwal ujian akhir semester ini digunakan metode algoritma genetika untuk membantu menyelesaikan permasalahan.

Untuk menyelesaikan masalah pembuatan jadwal ujian akhir semester ini digunakan *constraint* yang berlaku di jurusan Teknik Informatika Unesa. Beberapa komponen data terkait dengan pembuatan jadwal yaitu data matakuliah, peserta matakuliah, ruang serta waktu ujian. Faktor dosen tidak diikutkan dalam *constraint* karena dalam ujian akhir semester pengawas ujian tidak harus dosen yang bersangkutan. Untuk ujicoba digunakan data jadwal semester genap 2014/2015. Hasil ujicoba menunjukkan bahwa algoritma genetika dapat

digunakan untuk menyelesaikan masalah penjadwalan kuliah.

Kelemahan dalam pembuatan jadwal dengan algoritma genetika ini, jika diinginkan hasil yang cukup optimal maka diperlukan jumlah populasi dan generasi yang cukup besar. Tetapi jika dimasukkan jumlah populasi dan generasi yang besar maka proses penjadwalan menjadi sangat lama.

Saran

Berdasar hasil penelitian ini maka sebagai saran yang bisa digunakan untuk pengembangan penelitian ini bisa dicoba pembuatan jadwal dengan metode-metode yang lain untuk mendapatkan proses penjadwalan yang lebih cepat.

DAFTAR PUSTAKA

Asif Ansari, Prof Sachin Bojewar, 2014, Genetic Algorithm to Generate the Automatic Time-Table, International Journal on Recent and Innovation Trends in Computing and Communication, volume 2 Issue 11.

Leon Bambrick, 1997, Lecture Timetabling Using Genetic Algorithms.

Sri Kusumadewi, 2003, Artificial Intelligence(Teknik dan Aplikasinya). Graha Ilmu Yogyakarta

Suyanto, 2005, Algoritma Genetika dalam Matlab. Yogyakarta: Andi.

Suyanto, 2010, Algoritma Optimasi; Deterministik atau Probabilistik, Graha Ilmu Yogyakarta.

Zainudin Zuhri, 2014, Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi, Andi Yogyakarta

