KOMBINASI ALGORITME GENETIKA DAN TABU LIST PADA KASUS PENJADWALAN UJIAN

(Studi Kasus: Universitas Kristen Immanuel)

COMBINATION OF GENETIC ALGORITHM AND TABU LIST IN THE CASE OF EXAM SCHEDULING

(Case Study: Universitas Kristen Immanuel)

Yo'el Pieter Sumihar *[1], Aina Musdholifah, S.Kom., M.Kom., Ph.D.[*2]
[1]Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: ***[1]**yoel.pieter.s@mail.ugm.ac.id , [*2] aina_m@ugm.ac.id

Abstrak

Sistem penjadwalan ujian bertujuan untuk memberikan solusi untuk penjadwalan ujian yang selama ini masih dikerjakan secara manual. Beberapa penelitian mengenai penjadwalan sudah dilakukan, salah satunya dengan algoritme genetika. Namun masih ada kelemahan yang terjadi pada algoritme genetika yaitu seringnya terjebak dalam local optimum. Penelitian ini mengembangkan sistem penjadwalan menggunakan kombinasi algoritme genetika dan tabu list untuk mengatasi kelemahan sistem sebelumnya. Penelitian ini menggunakan data Universitas Kristen Immanuel pada program studi Teknik Informatika. Hasil penelitian algoritme genetik dan tabu list mendapatkan hasil bahwa algoritme genetik berhasil menghindari local optimun dengan menggunakan tabu list sebagai syarat agar nilai hasil operasi genetika tidak boleh digunakan jika hasil tersebut sudah ada di dalam tabu list. Proses penjadwalan menggunakan algoritme genetika dan tabu list mampu mengurangi generasi lebih banyak dibandingkan dengan algoritme genetika. Namun kelemahan yang dimiliki algoritme genetika dengan tabu list adalah waktu proses yang cenderung lebih lambat dibandingkan dengan algoritme genetika tanpa tabu list.

Kata kunci: penjadwalan, algoritme genetika, tabu list, jadwal ujian, tabu.

Abstract

The exam scheduling system aims to provide solutions for scheduling exams that have been done manually. Several studies on scheduling have been done, one with a genetic algorithm. But there are still weaknesses that occur in the genetic algorithm is often trapped in local optimum. This study developed a scheduling system using a combination of genetic algorithms and tabu lists to address the weaknesses of the previous system. This research uses Immanuel Christian University data on Informatics Engineering course. Genetic algorithm and taboo list results show that genetic algorithm succeeded in avoiding local optimum by using taboo list as a requirement that genetic result value should not be used if the result is already in taboo list. The process of scheduling using genetic algorithms and taboo lists can reduce the generation of more than the genetic algorithm. But the disadvantage of genetic algorithms with taboo lists is that process time tends to be slower than the genetic algorithm without taboo lists.

Keywords: scheduling, genetic algorithm, taboo list, scheduling exams, taboo.

1. Pendahuluan

Penjadwalan ujian adalah suatu kasus yang menantang bagi dunia komputasi. Solusi yang dihasilkan dalam proses penjadwalan bisa terhitung bahkan bisa tidak terhitung. Solusi yang dihasilkan belum tentu solusi optimum sehingga pencarian terhadap solusi optimum bisa memakan cukup waktu tergantung dari kasus penjadwalan yang terjadi. Oleh karena itu penjadwalan masuk dalam kategori permasalahan non-deterministic polynomial (NP)

dan kombinatorial (Kazarlis, 2005). Algoritme genetika (GA) bagus dalam mengambil ruang pencarian yang besar dan berpotensi besar serta mengarahkan untuk mencari kombinasi yang optimal dari berbagai hal dan solusi yang sulit dicapai. Algoritme genetika (GA) adalah teknik pencarian, optimisasi, dan pembelajaran mesin adaptif yang berulang yang didasarkan pada prinsip-prinsip seleksi alam, dan mampu menemukan solusi untuk masalah sulit NP (Panchal, 2015). Karena itu digunakan algoritme genetika untuk menyelesaikan kasus penjadwalan pada penelitian ini.

Salah satu masalah yang dihadapi ketika menggunakan Algoritme genetika adalah seringnya terjebak dalam local optimum yang mengakibatkan tidak didapatkannya semua kemungkinan yang ada. Maka dari itu penelitian ini akan mencoba menambahkan tabu list untuk menyimpan nilai optimal untuk menghindari dan mengevaluasi kandidat solusi yang pernah dikunjungi sebelumnya.

2. Metode Penelitian

Algoritme Genetika

Algoritme genetika dimulai dengan pengaturan parameter yang diperlukan dalam algoritme genetika. Parameter yang diatur meliputi panjang kromosom (*g*), banyak anggota populasi (*p*), banyaknya generasi atau iterasi *(i),* fungsi *fitness f(k),* probabilitas mutasi (p_m), dan probabilitas *crossover* (p_c)*.* Setelah itu dilanjutkan dengan proses inisiasi populasi awal yang memiliki panjang *g* gen. Setiap gen berisi jadwal ujian secara utuh.

Populasi yang telah dibentuk akan dikenakan proses evaluasi untuk menghitung nilai *fitness* setiap kromosom dengan menggunakan *fitness f(k)* yang sebelumnya telah diatur, kemudian dilanjutkan dengan proses pemeriksaan parameter penghentian proses. Parameter yang digunakan sebagai syarat penghentian proses adalah banyaknya generasi atau iterasi *(i)* dan nilai dari *fitness*. Jika parameter belum terpenuhi, maka proses seleksi akan terus berulang. Seleksi orangtua dilakukan dengan seleksi dengan *elitism*e untuk memilih kromosom sebagai orangtua *(parents)* pada *tabu list*. *Crossover* dan mutasi dilakukan dengan mereferensi kepada probabilitas *crossover* (p_c) dan probabilitas mutasi (p_m). Setelah itu, dihasilkan kromosom baru (*offspring*) dari operasi genetika yang kemudian dilanjutkan dengan proses *update* generasi dengan populasi yang baru. Populasi baru untuk *update* generasi menggunakan *elitisme* untuk menjaga agar kromosom dengan nilai *fitness* yang baik tidak hilang dari populasi dan masuk ke generasi selanjutnya. Namun sebelum proses untuk mendapatkan generasi selanjutnya dapat dilanjutkan, hasil dari proses algoritme genetika tersebut akan dikenakan proses pengecekan ke dalam *tabu list* terlebih dahulu. Jika hasil dari operasi genetika sudah ada dalam *tabu list*, maka proses algoritme genetika harus diulangi hingga menghasilkan hasil yang belum terdapat pada *tabu list*, dan hasil yang belum terdapat pada *tabu list* tersebut akan dimasukkan ke dalam *tabu list* untuk digunakan sebagai parameter pada generasi berikutnya.

Proses sistem penjadwalan ujian dengan algoritme genetika dan tabu list adalah sebagai berikut:

1 Pengaturan Parameter

Pengaturan parameter algoritme genetika meliputi cacah anggota populasi *(p),* cacah generasi atau iterasi *(i),* probabilitas *crossover* (p_c) dan probabilitas mutasi (p_m). Panjang kromosom *(g)* yang digunakan disesuaikan dengan banyaknya mata kuliah yang akan diujikan, pada penelitian ini panjang kromosom yang akan digunakan adalah 53 karena menyesuaikan dengan jumlah mata kuliah. Cacah anggota populasi minimal sama dengan

ukuran kromosom. Nilai parameter yang akan digunakan akan diubah pada saat proses percobaan, yang diharapkan agar mendapat hasil yang lebih maksimal.

2 Representasi Kromosom

Representasi kromosom diperlukan untuk menyajikan alternatif solusi kromosom. Pada penelitian ini alternatif solusi yang digunakan berupa jadwal ujian, sehingga kromosom direpresentasikan sebagai jadwal ujian matakuliah yang akan diujikan. Pada representasi ini maka panjang dari kromosom akan sama dengan banyaknya kelas (*k*) yang akan diujikan.

Setiap gen pada kromosom merepresentasikan kelas yang akan diujiakan. Kelas yang akan diujikan berisi kombinasi dari mata kuliah, ruang kelas dan waktu ujian. P setiap kode matakuliah berisi nama matakuliah, kode matakuliah, semester, jenis ruangan, jenis matakuliah (eksak dan non eksak), jumlah siswa dan dosen yang mengajar, data-data inilah yang akan direpresentasikan sebagai kromosom. Untuk desain dari sebuah gen yang membentuk kromosom, dapat dilihat pada Gambar 1.

Gambar 1 Desain sebuah Gen

Pada Gambar 1 dapat dilihat bahwa desain representasi kelas oleh gen berisi tiga parameter, berikut penjelasan dari masing-masing parameter:

a. Kodekelas: adalah kode kelas yang akan diujikan, kodeKelas sendiri akan diinputkan secara berurutan untuk menghindari kesalahan pada proses crossover dan mutasi.
b. kodeRuangan: adalah kode ruangan yang akan digunakan pada saat ujian.
c. kodeWaktuUjian: adalah kode waktu ujian yang digunakan pada saat ujian.

Setiap gen akan diinputkan berdasarkan data yang sudah diperoleh dari sumber data, contoh gen dapat dilihat pada Gambar 2

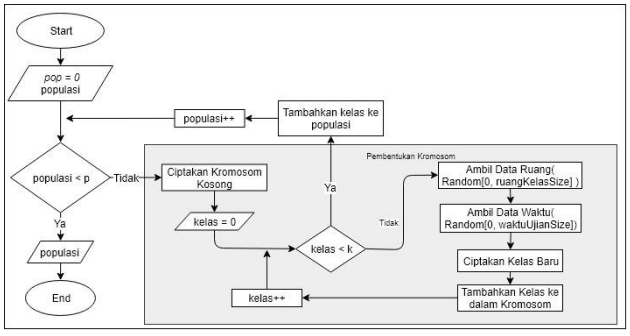
Gambar 2 Contoh representasi sebuah Gen

Setiap gen yang berkumpul akan membentuk sebuat kromosom (individu), contoh dari representasi kromosom dapat dilihat pada Gambar 3.

Gambar 3 Contoh representasi kromosom

3 Inisialisasi populasi

Inisialisasi sejumlah *p* populasi dari setiap kromosom berukuran *k*, dimana *k* merupakan banyaknya kelas yang akan diujikan. Diagram alur proses inisialisasi popululasi dapat dilihat pada Gambar 4.



Gambar 4 Diagram alur proses inisialisasi populasi

Pada proses pembentukan kromosom, setiap gen akan diciptakan secara berurutan berdasarkan kode matakuliah yang akan diujikan, digabungkan dengan data ruang ujian dan waktu ujian yang diciptakan secara acak dengan cara mengambil data secara acak dari list data ruang ujian maupun data waktu ujian. Matakuliah diciptakan secara berurutan agar tidak terjadi kesalahan pada proses *crossover* dan mutasi, dimana proses *crossover* dan mutasi dilakukan pada matakuliah yang sama. Contoh hasil pembentukan populasi *p* pada kelas *k* dapat dilihat pada Gambar 5.

Kromosom 1	C1, PUSKOM, MT3	C2, IV.B.I, MT4	C3, IV.B.I, MT5	C4, IV.B.I, MT6
Kromosom 2	C1, IV.A.III, MT4	C2, PUSKOM, MT5	C3,PUSKOM, MT5	C4, IV.B.I, MT7
Kromosom 3	C1, IV.A.II, MT15	C2, IV.A.I, MT16	C3, IV.B.I, MT7	C4, IV.A.I, MT12
Kromosom 4	C1, PUSKOM, MT4	C2, IV.B.I, MT5	C3, IV.B.I, MT6	C4, IV.B.I, MT8
Kromosom 5	C1, IV.A.III, MT5	C2, PUSKOM, MT6	C3,PUSKOM, MT6	C4, IV.B.I, MT9
Kromosom 6	C1, IV.A.II, MT16	C2, IV.A.I, MT17	C3, IV.B.I, MT8	C4, IV.A.I, MT13
Kromosom 7	C1, PUSKOM, MT5	C2, IV.B.I, MT6	C3, IV.B.I, MT7	C4, IV.B.I, MT10

Gambar 5 Contoh inisialisasi populasi awal

4 Evaluasi

Setelah tahap pembangkitan populasi, tahap selanjutnya adalah tahap evaluasi individu. Tahap ini bertujuan untuk mengetahui kualitas (tingkat keoptimalan) dari setiap kromosom dalam suatu populasi. Sehingga dapat diketahui kromosom mana yang merupakan kromosom terbaik (paling optimal). Kromosom yang paling optimallah yang akan dipilih sebagai solusi dari permasalahan ini.

Proses evaluasi dilakukan untuk mencari nilai *fitness*. Diagram alur proses evaluasi terdapat pada Gambar 6. Pada gambar tersebut dijelaskan proses detail dari evaluasi pada algoritme genetika dalam sistem penjadwalan ujian. Proses evaluasi dimulai dengan menginisialisasi kromosom pertama dengan indeks 0. Selanjutnya dilanjutkan dengan perulangan dengan menambah nilai indeks sebanyak nilai iterasi yang sudah ditentukan sebelumnya. Perhitungan nilai *fitness* menggunakan fungsi *fitness* yang sudah ditentukan dengan menggunakan aturan pada *Hard constraint* dan *Soft constraint* yang sudah ditentukan, dimana setiap *constraint* memiliki nilai bobot. Penanganan kendala (*constraint*) pada penelitian ini menggunakan strategi pemberian penalti (*penalizing strategy*). Adapun batasan keras (*hard contraint*) dan batasan lunak (*soft constraint*) pada sistem penjadwalan ini adalah :

Batasan Keras (*Hard constraint*) :

a. Tidak boleh ada jadwal ujian yang menggunakan ruangan yang sama dalam waktu yang sama
b. Tidak boleh ada ujian yang bersamaan dalam semester yang sama
c. Tidak boleh ada ujian yang bersamaan yang diambil oleh mahasiswa yang sama
d. Tidak boleh ada ujian dari dosen yang sama di hari, jam yang sama.

Batasan Lunak (*Soft constraint*) :

a. Kelas kecil akan diusahakan untuk dimasukkan dalam ruangan yang memiliki daya tampung yang kecil pula.
b. Mata kuliah memiliki jenis golongan eksak dan teori, golongan eksak diusahakan untuk diujikan di pagi hari (sesi 1).
c. Ujian praktikum harus dilaksanakan di lab praktikum
d. Hari Sabtu mungkin dipakai tapi hanya sebagai alternatif.

Batasan Keras dapat digunakan sebagai aturan *fitness* dengan nilai bobot yang lebih besar dari batasan lunak. Selanjutnya ditentukan urutan dan nilai prioritasnya dari batasan di atas. Rumus fungsi *fitness* kromosom k dapat didefinisikan sebagai persamaan 1.

$$f(k) = \frac{1}{(B+0.1)} \quad (1)$$

Keterangan : $B = \sum_{i=1}^{n} J_i$

$J_i = Bobot\ batasan\ ke - i\ yang\ dilanggar$

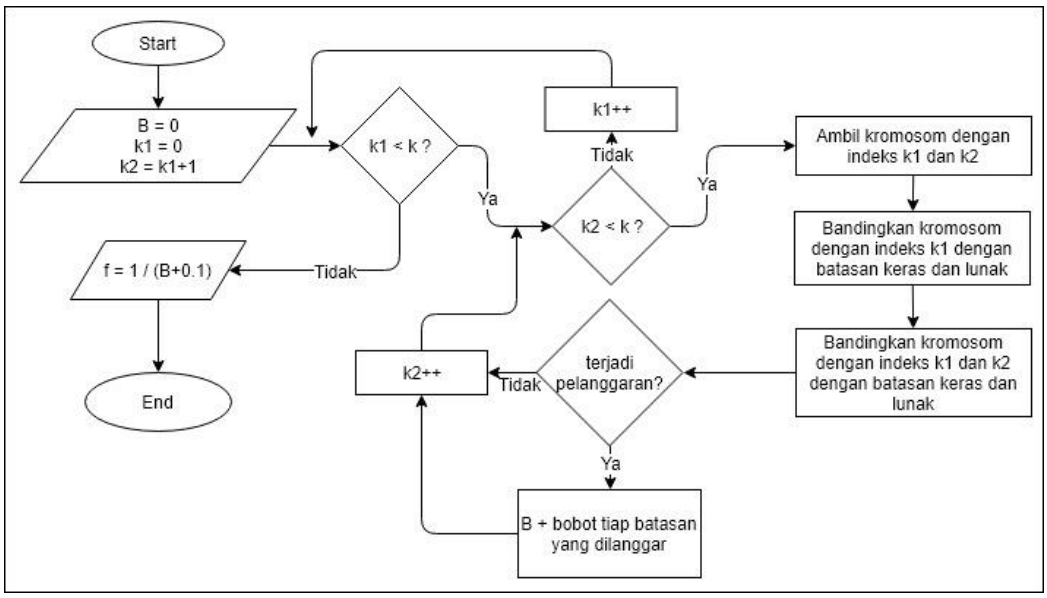
$n = cacah\ dari\ batasan\ yang\ dilanggar$

0.1 *adalah konstanta untuk menghindari pembagian dengan nilai* 0

Pemberian bobot pada setiap *constraint* dapat dilihat pada Tabel 1. Setiap kromosom akan dievaluasi menggunakan fungsi *fitness* pada persamaan 1 dengan bobot pada setiap konfliknya berdasarkan pada Tabel 1 Diagram alur proses evaluasi dapat dilihat pada Gambar 6.

Tabel 1 Bobot Batasan Keras dan Batasan Lunak

Aturan	Bobot
Hard constraint point a	5
Hard constraint point b	5
Hard constraint point c	5
Hard constraint point d	4
Soft constraint point a	2
Soft constraint point b	1
Soft constraint point c	2
Soft constraint point d	3



Gambar 6 Diagram Alir Proses Evaluasi

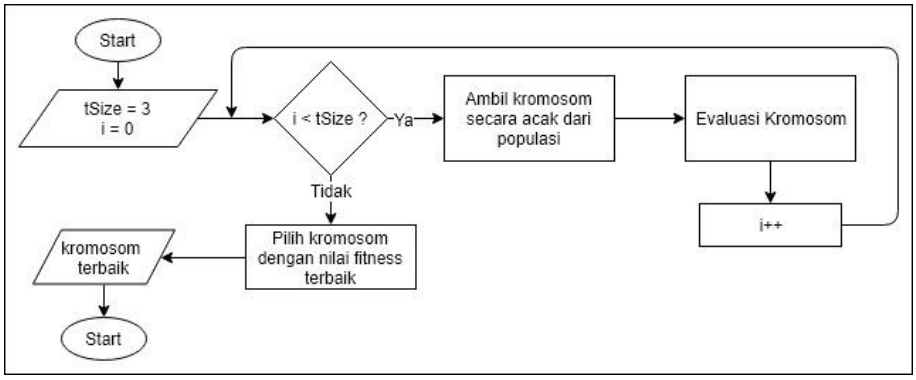
Proses evaluasi dilakukan pada setiap kromosom dengan cara memeriksa apakah ada pelanggaran yang terjadi pada kromsom tersebut. Pelanggaran bisa saja pada kromosom itu sendiri dan perbandingan pada kromosom lain. Maka dari itu pada diagram alur terdapat proses membandingkan antara kromosom pada indeks k1 dan k2 untuk memeriksa apakah ada pelanggaran yang terjadi seperti pelanggaran pada *hard constraint* sedangkan pemeriksaan tanpa perbandingan dengan kromosom lain digunakan untuk pemeriksaan pada *soft constraint*. Ilustrasi dari hasil evaluasi dapat dilihat pada Gambar 7.

Kromosom					Constraint		nilai
					Hard	Soft	fitness
Kromosom 1	C1, PUSKOM, MT3	C2, IV.B.I, MT4	C3, IV.B.I, MT5	C4, IV.B.I, MT6	a	b,c	0.140845
Kromosom 2	C1, IV.A.III, MT4	C2, PUSKOM, MT5	C3,PUSKOM, MT5	C4, IV.B.I, MT7	a,b	c	0.082645
Kromosom 3	C1, IV.A.II, MT15	C2, IV.A.I, MT16	C3, IV.B.I, MT7	C4, IV.A.I, MT12	-	b	0.47619
Kromosom 4	C1, PUSKOM, MT4	C2, IV.B.I, MT5	C3, IV.B.I, MT6	C4, IV.B.I, MT8	a	b,c	0.140845
Kromosom 5	C1, IV.A.III, MT5	C2, PUSKOM, MT6	C3,PUSKOM, MT6	C4, IV.B.I, MT9	a,b	c	0.082645
Kromosom 6	C1, IV.A.II, MT16	C2, IV.A.I, MT17	C3, IV.B.I, MT8	C4, IV.A.I, MT13	a	b,c	0.140845
Kromosom 7	C1, PUSKOM, MT5	C2, IV.B.I, MT6	C3, IV.B.I, MT7	C4, IV.B.I, MT10	a,b	c	0.082645

Gambar 7 Ilustrasi hasil evaluasi kromosom

5 Seleksi

Seleksi kromosom menggunakan seleksi dengan metode *tournament selection*, dimana pada proses seleksi ini akan mengambil kromosom secara acak, lalu diadu untuk mendapatkan kromosom dengan nilai *fitness* terbaik saja. Diagram alir proses seleksi dapat dilihat pada Gambar 8.



Gambar 8 Diagram alir tournament selection

Seleksi dengan *tournament selection* dimulai dengan menentukan ukuran dari banyaknya kromosom yang akan diadu. Selanjutnya dilakukan proses evaluasi pada kromosom tersebut untuk mendapatkan nilai *fitness* dari masing-masing kromosom, kromosom dengan nilai *fitness* terbaiklah yang akan digunakan pada proses ini. Ilustrasi *tournament selection* dapat dilihat pada Gambar 9.

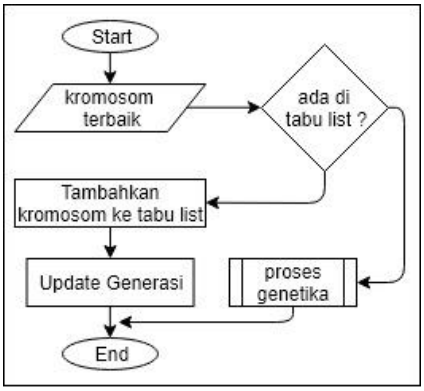
Kromosom 1	C1, PUSKOM, MT3	C2, IV.B.I, MT4	C3, IV.B.I, MT5	C4, IV.B.I, MT6	
Kromosom 2	C1, IV.A.III, MT4	C2, PUSKOM, MT5	C3,PUSKOM, MT5	C4, IV.B.I, MT7	
Kromosom 3	C1, IV.A.II, MT15	C2, IV.A.I, MT16	C3, IV.B.I, MT7	C4, IV.A.I, MT12	
Kromosom 4	C1, PUSKOM, MT4	C2, IV.B.I, MT5	C3, IV.B.I, MT6	C4, IV.B.I, MT8	
Kromosom 5	C1, IV.A.III, MT5	C2, PUSKOM, MT6	C3,PUSKOM, MT6	C4, IV.B.I, MT9	
Kromosom 6	C1, IV.A.II, MT16	C2, IV.A.I, MT17	C3, IV.B.I, MT8	C4, IV.A.I, MT13	
Kromosom 7	C1, PUSKOM, MT5	C2, IV.B.I, MT6	C3, IV.B.I, MT7	C4, IV.B.I, MT10	
Kromosom 2	C1, IV.A.III, MT4	C2, PUSKOM, MT5	C3,PUSKOM, MT5	C4, IV.B.I, MT7	*f= 0.12*
Kromosom 4	C1, PUSKOM, MT4	C2, IV.B.I, MT5	C3, IV.B.I, MT6	C4, IV.B.I, MT8	*f= 0.25*
Kromosom 7	C1, PUSKOM, MT5	C2, IV.B.I, MT6	C3, IV.B.I, MT7	C4, IV.B.I, MT10	*f=0.32*
Kromosom 7	C1, PUSKOM, MT5	C2, IV.B.I, MT6	C3, IV.B.I, MT7	C4, IV.B.I, MT10	*f=0.32*

Gambar 9 Ilustrasi tournament selection

6 Tabu List

Pada proses seleksi sebelumnya didapatkan kromosom terbaik, namun sebelum dilakukannya update generasi, maka dilakukan proses pengecekan ke dalam tabu list, jika hasil yang sama sudah terdapat di dalam *tabu list*, maka proses genetik algoritma akan diulangi kembali sampai menemukan hasil yang belum terdapat di dalam *tabu list* untuk

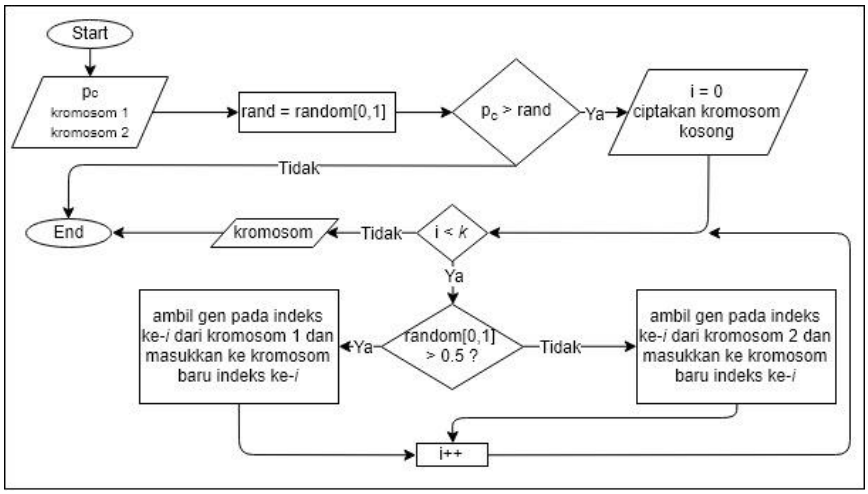
menghindari *local optimum*. Diagram alir proses pengecekan ke dalam *tabu list* dapat dilihat pada Gambar 10



Gambar 10 Diagram alur *tabu list*

7 Crossover

Kromosom baru akan terbentuk dari hasil *crossover* dan mutasi. Akan dipilih dua orangtua (*parents*) secara acak dari kromosom terbaik, dan kromosom sudah terpilih akan dikenakan proses *crossover*. Teknik *uniform crossover* digunkan untuk menghasilkan *offspring*, pada dasarnya teknik ini tidak membagi kromosom ke dalam segmen, tetapi memperlakukan setiap gen secara terpisah. Diagram alir proses *crossover* dapat dilihat pada Gambar 11.



Gambar 11 Diagram alir proses uniform crossover

parent

C1, PUSKOM, MT3	C2, IV.B.I, MT4	C3, IV.B.I, MT5	C4, IV.B.I, MT6	C5, IV.B.I, MT7
C1, IV.A.III, MT4	C2, PUSKOM, MT5	C3,PUSKOM, MT5	C4, IV.B.I, MT7	C5, IV.B.I, MT8

Offspring

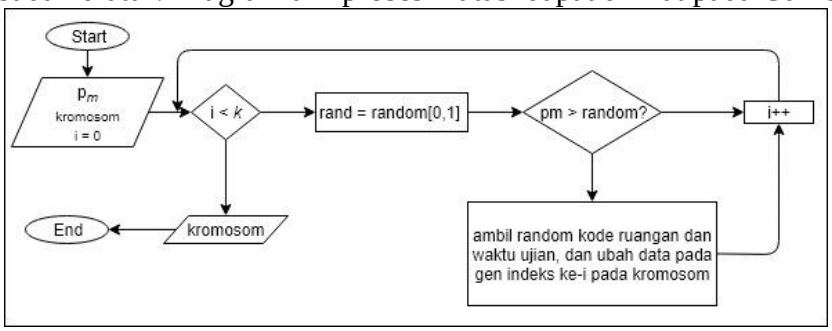
C1, PUSKOM, MT3	C2, PUSKOM, MT5	C3, IV.B.I, MT5	C4, IV.B.I, MT6	C5, IV.B.I, MT8

Gambar 12 Contoh *uniform crossover* pada kromosom

Proses *crosssover* dilakukan berdasarkan probabilitas dari *crossover* tersebut. Pada proses *crossover* ini diumpamakan seperti melemparkan koin untuk setiap gen pada kromosom, untuk menentukan gen dari kromosom mana yang akan dimasukkan sebagai *offspring*. Ilustrasi proses *uniform crossover* digambarkan pada Gambar 12. Gambar 12 menjelaskan 2 kromsom *parents* yang disilangkan pada setiap gen akan menghasilkan sebuah *offspring*.

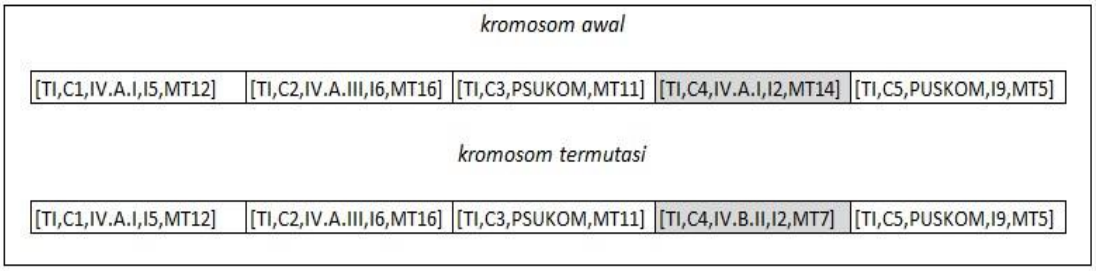
8 Mutasi

Teknik mutasi yang digunakan adalah pemilihan nilai secara acak untuk mengisi gen yang termutasi. Mutasi pada gen di dalam kromosom diproses berdasarkan probabilitas mutasi yang sudah diatur. Diagram alir proses mutasi dapat dilihat pada Gambar 13.



Gambar 13 Diagram alur pada proses mutasi

Pada Gambar 14, pada segmen ke 4 terjadi mutasi sehingga nilai dalam gen diubah secara acak seperti yang terlihat pada kromosom termutasi, namun nilai yang diubah hanya pada bagian ruangan dan waktu ujian.



Gambar 14 Contoh mutasi pemilihan nilai secara acak

9 Update Generasi

Pergantian populasi menggunakan *generational* model dengan elitisme. Seluruh anggota akan dihapus kecuali kromosom anggota populasi sebelumnya yang memiliki *fitness* tertinggi. Anggota populasi lain akan diisi oleh kromosom baru hasil *crossover* dan mutasi (*offspring*). Banyaknya anggota sebelumnya yang disimpan ditentukan menggunakan parameter yang ditentukan di awal.

10 Representasi kromosom terbaik.

Kromosom terbaik adalah kromosom dengan nilai *fitness* terbaik, kromosom ini akan digunakan sebagai jadwal ujian. Kromosom yang terdiri dari *k* gen yang setiap gen menyatakan kelas setiap matakuliah yang diujikan baik itu data mata kuliah, dosen yang mengampu matakuliah, ruangan yang dipakai ujian, hari dan jam ujian. Ilustrasi hasil ujian dapat dilihat pada Gambar 15.

No	Kode	Nama	Semester	Jenis	Room[kapasitas]	Dosen	Hari	Jam	Minggu Ke
1	C21	Metematika Kelas 1	1	Eksak	IV.B.I(45)	Emerita	Senin	08:00 - 10:30	(Week I)
2	C34	SPK	5	Non Eksak	PUSKOM(60)	Pak Anto	Senin	08:00 - 10:30	(Week I)
3	C38	PSD	3	Eksak	IV.A.I(45)	Jatmika	Senin	08:00 - 10:30	(Week I)
4	C11	PTI Kelas 3	1	Non Eksak	PUSKOM(60)	Pieter	Senin	12:00 - 14:00	(Week I)
5	C40	P. Agama Kelas 2	1	Non Eksak	IV.B.I(45)	Pak Tertius S.	Senin	14:00 - 16:00	(Week I)
6	C18	Metnum Kelas 2	3	Eksak	PUSKOM(60)	Ibu Heny	Selasa	08:00 - 10:30	(Week I)
7	C53	PAK Kelas 3	1	Non Eksak	IV.A.II(45)	Dino	Selasa	08:00 - 10:30	(Week I)
8	C5	Pemrog Script	5	Non Eksak	PUSKOM(60)	Dino	Selasa	14:00 - 16:00	(Week I)

Gambar 15 Contoh hasil jadwal ujian

3. Hasil dan Pembahasan

3.1 Mencari Parameter Terbaik Pada Algoritme Genetika

Pada pengujian ini ditujukan untuk mendapatkan parameter terbaik pada algoritme genetika. Seperti yang telah dijelaskan sebelumnya bahwa setiap kombinasi antara probabilitas *crossover* dan probabilitas mutasi diujikan sebanyak 10 kali, sehingga pada Tabel 2 didapatkan hasil uji dengan nilai rata-rata *fitness* dan rata-rata waktu yang dibutuhkan untuk melakukan proses algoritme genetika.

Berdasarkan hasil uji yang dilakukan pada jadwal ujian menggunakan algoritme genetika dengan kombinasi antara probabilitas mutasi dan probabilitas *crossover*, maka didapatkan nilai terbaik *fitness* berada pada kombinasi $p_m = 0.01$ dan $p_c = 0.8$ dengan nilai *fitness* sebesar 0.283241542. Berdasarkan hasil pengujian pada Tabel 6.1 juga didapatkan waktu uji tercepat yang juga berada pada kombinasi $p_m = 0.01$ dan $p_c = 0.8$ dengan nilai waktu sebesar 5580.4 milidetik. Maka parameter terbaik pada algoritme genetik berada pada kombinasi antara $p_m = 0.01$ dan $p_c = 0.8$.

3.2 Mencari Parameter Terbaik Pada Algoritme Genetika Dan *Tabu List*

Pada pengujian ini ditujukan untuk mendapatkan parameter terbaik pada algoritme genetika dan *tabu list*. Seperti yang telah dijelaskan sebelumnya bahwa setiap kombinasi antara probabilitas *crossover* dan probabilitas mutasi diujikan sebanyak 10 kali, sehingga pada Tabel 3 didapatkan hasil uji dengan nilai rata-rata *fitness* dan rata-rata waktu yang dibutuhkan untuk melakukan proses algoritme genetika dan *tabu list*.

Berdasarkan hasil uji yang dilakukan pada jadwal ujian menggunakan kombinasi algoritme genetika dan *tabu list* dengan kombinasi antara probabilitas mutasi dan probabilitas *crossover*, maka didapatkan nilai terbaik *fitness* berada pada kombinasi $p_m = 0.01$ dan $p_c = 0.89$, $p_m = 0.015$ dan $p_c = 0.83$, $p_m = 0.025$ dan $p_c = 0.84$ dengan nilai *fitness* sebesar 0.275373721. Berdasarkan hasil pengujian pada Tabel 6.1 juga di dapatkan waktu uji tercepat berdasarkan nilai *fitness* terbesar yang berada pada kombinasi $p_m = 0.015$ dan $p_c = 0.83$ dengan nilai waktu sebesar 7832.5 milidetik. Maka parameter terbaik pada algoritme genetik berada pada kombinasi antara $p_m = 0.015$ dan $p_c = 0.83$.

3.3 Perbandingan banyaknya generasi dan waktu

Pada pengujian ini akan berfokus pada banyaknya generasi dan waktu yang dibutuhkan untuk melakukan proses pembuatan sistem ujian menggunakan kedua algoritme yang dibandingkan. Pertama pengujian akan dilakukan pada parameter terbaik dari algoritme genetika. Masing-masing pengujian dilakukan sebanyak 10 kali untuk mendapatkan nilai terbaik. Hasil pengujian dapat dilihat pada Tabel 2.

Pada hasil penelitian pada Tabel 2 dapat dilihat bahwa hasil *fitness* dan waktu dari algoritme genetika lebih baik dari hasil *fitness* pada algoritme genetika dan *tabu list* pada parameter yang sama, namun pada algoritme genetika dan *tabu list* lebih unggul dalam

banyaknya generasi karna algoritme genetika dan *tabu list* hanya membutuhkan 38 generasi sedangkan algoritme genetika selalu lebih dari 300 generasi.

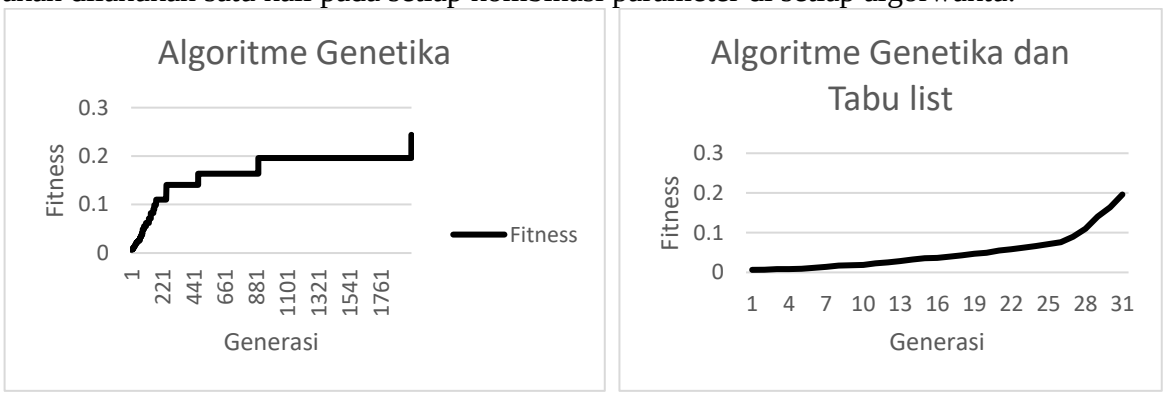
Tabel 2 Perbandingan Banyaknya generasi dan waktu

No	Algoritme	Pm	Pc	Fitness	Waktu	Generasi
1	Genetika	0.01	0.8	0.283	5580.4	315
2	Genetika dan Tabu list	0.01	0.8	0.267	7343.5	38
3	Genetika	0.015	0.83	0.260	6843.4	341
4	Genetika dan Tabu list	0.015	0.83	0.275	7832.5	38

Pada hasil penelitian dalam Tabel 2 juga dapat dilihat bahwa dari segi waktu, algoritme genetika masih lebih baik dari pada kombinasi algoritme genetika dan *tabu list* pada masing-masing skema parameter. Waktu terbaik dari algoritme genetika adalah 5580.4 milidetik, sedangkan waktu terbaik dari kombinasi algoritme genetika dan *tabu list* sebesar 7832.5 milidetik.

3.4 Perbandingan Nilai *Fitness* Pada Tiap Generasi

Pada pengujian ini telah dilakukan pegujian untu k mendapatkan nilai *fitness* pada setiap generasi. Pengujian akan membandingkan algoritme genetika dengan algoritme genetika dan *tabu list* berdasarkan parameter terbaik masing-masing algoritme. Pengujian akan dilakukan satu kali pada setiap kombinasi parameter di setiap algorwaktu.

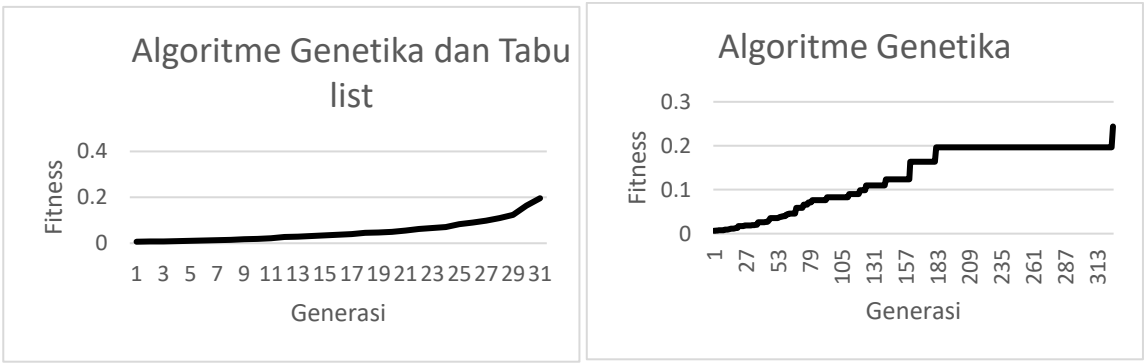


Gambar 16 Grafik *Fitness* Pada Tiap Generasi Algoritme Genetika dan Kombinasi Algoritme Genetika dengan *Tabu list***.**

Pengujian pertama membandingkan nilai *fitness* pada tiap generasi menggunakan nilai parameter terbaik dari algoritme genetika dengan nilai pm=0.01 dan nilai pc = 0.8. Hasil pengujian dapat di lihat pada Gambar 16. Pada hasil pengujian pada Gambar 16 dapat dilihat bahwa hasil pengujian pada algoritme genetika banyak nilai *fitness* yang berulang sebelum akhirnya mendapatkan nilai *fitness* yang lebih baik. Sedangkan pada algoritme genetika dan *tabu list* nilai *fitness* pada tiap generasi selalu berbeda dan nilainya selalu naik dikarenakan pengecekan hasil operasi genetika yang diperiksa ke dalam *tabu list* sehingga nilai yang sama tidak akan digunakan sebagai *parents*.

Pengujian kedua membandingkan nilai *fitness* pada tiap generasi menggunakan nilai parameter terbaik dari algoritme genetika dan *tabu list* dengan nilai pm=0,015 dan nilai pc = 0,83. Hasil pengujian dapat dilihat pada Gambar 16 dan Gambar 17.

Pada hasil pengujian yang dapat dilihat pada Gambar 17 dapat dilihat dari hasil pengujian pada algoritme genetika bahwa banyak nilai *fitness* yang berulang sebelum akhirnya mendapatkan nilai *fitness* yang lebih baik. Sedangkan pada algoritme genetika dan *tabu list* nilai *fitness* pada tiap generasi selalu berbeda dan nilainya selalu naik dikarenakan pengecekan hasil operasi genetika yang diperiksa ke dalam *tabu list* sehingga nilai yang sama tidak akan digunakan sebagai *parents*.



Gambar 17 Grafik *Fitness* Pada Tiap Generasi Algoritme Genetika.

4. Kesimpulan

Berdasarkan penelitian dan hasil pengujian, maka didapatkan kesimpulan yaitu: (1) Kombinasi algoritme genetika dan *tabu list* dapat digunakan untuk menghasilkan penjadwalan ujian di Universitas Kristen Immanuel, dan mampu menghasilkan nilai *fitness* yang lebih baik pada setiap generasinya. (2) Kombinasi algoritme genetika dan *tabu list* dapat menghindari *local optimum* yang terjadi algoritme genetika. (3) Kombinasi algoritme genetika dan *tabu list* berjalan lebih lambat jika dibandingkan dengan algoritme genetika. (4) Keunggulan kombinasi algoritme genetika dengan *tabu list* terletak pada mampunya algoritme ini untuk menghindari penggunaan nilai *fitness* yang sama sehingga dapat menghindari generasi yang terlalu banyak. Berdasarkan parameter terbaik algoritme genetika dan *tabu list* hanya membutuhkan 38 generasi untuk mendapatkan hasil terbaik, sedangkan algoritme genetika pada parameter terbaiknya membutuhkan 315 generasi untuk mendapatkan hasil yang terbaik.

Daftar Pustaka

Abramson, D., dan Abela, J., 1992, A Parallel Genetic Algoritm for Solving The School Timetabling Problem 15, *Australian Computer Science Conference (ASCSC)*, 29 - 31 Januari, pp. 1-11.

Albar, M. A., 2013, Algoritma Genetik Tabu Search Dan Algoritma Memetika Pada Permasalahan Penjadwalan Kuliah, *Tesis*, Universitas Gadjah Mada, Yogyakarta.

Amaral, P., dan Cardal, T., 2016, Computers & Operations Research Compromise ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling. *Computers and Operation Research*, 72, 160–174.

Bambrick, L., 1997, Lecture Timetabling Using Genetic Algoritms, *Thesis*, University of Queensland.

Beligiannis,G. N., Moschopoulos, C., dan Likothanassis, S.D., 2009, A Genetic Algortim Approach To School Timetabling, *Journal of the Operational Research Society*, 60, pp. 23-42.

Chorbev, I., Loskova, S., Dimitivski, I., dan Mihajlov, D., 2008, Solving the Highschool schedulling problem modelled with constraints satisfaction using Hybrid Heuristic Algoritms, Greedy Algoritms, Witold Bednorz(Ed.), *ISBN* : 978-953-7619-27-5, InTech.

Erama, R. 2013, Modifikasi Algoritma Genetika untuk penyelesaian permasalah penjadwalan pelajaran sekolah (Studi Kasus : SMPN 1 Telaga Gorontalo), *Tesis*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.

Glover, F., dan Laguna, M., 1997, *Tabu Search*, Kluwer Academic Publishers, Boston,.

Golub, M., dan Jakobovi, D., 2009, *Exam Timetabling Using Genetic Algorithm*, 357–362.

Jaengchuea, S., 2015, *A Hybrid Genetic Algorithm with Local Search and Tabu Search Approaches for Solving the Post Enrolment Based Course Timetabling Problem : Outperforming Guided Search Genetic Algorithm*, 29–34.

Jha, S.K., 2014, *Exam timetabling problem using genetic algorithm*, 649-654.

Kusumadewi, S., 2003, *Artificial Intelligent*, Graha Ilmu, Yogyakarta.

Kusumadewi, S., dan Purnomo, H., 2005 *Penyelesaian Masalah optimasi dengan teknik-teknik heuristik*, Graha Ilmu, Yogyakarta .

Mitchell,G.G., O'Donoghue, D., Barnes, D., McCarville, M., 2003, *Gene Repair-A Repair Operator for Genetic Operator for Genetic Algoritms*, Department of Computer Science, National University of Ireland Maynooth, Co. Kildare, Ireland, georgem@cs.may.ie.

Mutakhiroh, I., Saptono, F., Hasanah, N., dan Wiryadinata, R., 2007, Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut Dan Algortima Genetika, *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta.

Pillay, N., dan Banzhaf, W., 2010, An Informed genetic algoritm for the examination timetabling problem, *Applied Soft Computing*, Vol.10, No.2, Pp. 457-467.

Putra, D. M. D. U., 2012, Penerapan Algortima Genetika Untuk Menyelesaikan Permasalahan Penjadwalan Perawat Dengan Fuzzy Fitness Function, *Tesis*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.

Raghavjee, R., dan Pillay, N., 2010, Using Genetic Algoritms to solve the south African school timetabling problem, *Nature and Biologically Inspired Computing (NaBIC)*, 15 - 17 Desember, pp. 286-292.

Sahputra, I., 2016, Optimasi Pembentukan Kelas Belajar Siswa Menggunakan Hybrid Genetic Algorithm, *Tesis*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.

Syarif, D. E., 2014, *Algoritma Genetika Teori dan Aplikasi*. Yogyakarta: Graha Ilmu.

Zdansky, M., dan Pozivil, J., Combination genentic/tabu search algorithm for hybrid flowshops optimization, *Tesis*, Department of computing and control engineering, Faculty of chemical engineering, Prague Institute of Chemical Technology, Technicka, Czech Republic, 2002.