

Optimasi Penjadwalan Damping Mahasiswa Difabel Menggunakan Algoritma Genetika (Studi Kasus PSLD Universitas Brawijaya)

M. Mart Hans Luber¹, Imam Cholissodin², Candra Dewi³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹mart.hans.luber@gmail.com, ²imamcs@ub.ac.id, ³dewi_candra@ub.ac.id

Abstrak

Penjadwalan pendampingan mahasiswa difabel merupakan penyusunan jadwal pelaksanaan seorang pendamping yang bertugas serta dalam pembagian waktu kerja. Pada proses penjadwalan yang baik maka akan memaksimalkan pelayanan terhadap mahasiswa difabel. Permasalahan penjadwalan ini menjadi hal yang sulit karena jumlah pendamping yang relative terbatas dibanding dengan jumlah mahasiswa difabel. Jadwal dibuat dengan beban kerja merata untuk setiap pendamping. Dalam penelitian ini diterapkan konsep penyelesaian masalah penjadwalan dengan menggunakan algoritma genetika. Penerapan algoritma genetika untuk mencari solusi optimal. Dalam penyelesaian masalah ini digunakan representasi integer dengan panjang kromosom 45 gen yang setiap bagian gen menunjukkan kode pendampingan. Metode *crossover* yang digunakan adalah metode one-cut point, proses mutasi menggunakan metode reciprocal exchange *mutation* dan pada proses seleksi menggunakan metode elitism selection. Dari hasil pengujian yang telah dilakukan diperoleh parameter optimal yaitu menggunakan 100 generasi dengan nilai *fitness* 0.966. Hasil akhir yang didapat berupa jadwal pendampingan selama 5 hari.

Kata kunci: algoritma genetika, penjadwalan, mahasiswa difabel, pendamping

Abstract

Mentoring scheduling for Disabled student is the preparation schedule implementation to a companion who served in the division of working time. On the good scheduling process then will maximize service to disabled students. This scheduling problem is difficult because the number of accompanying relative limited compared to the number of disabled students. The schedule created by the workload evenly to each escort. In this study applied the concepts of problem solving scheduling by using genetic algorithms. Application of genetic algorithm to find the optimal solution. In the settlement of this problem use an integer representation with the length of chromosome 45 genes that each section of the gene showed code mentoring. The method used is the crossover one-cut method of point mutation process, using the method of reciprocal exchange and mutation on the selection process using the method of elitism selection. From the results of testing that has been done optimal parameters obtained using a 100 generation with fitness value 0.966. The final results obtained in the form of a mentoring schedule for 5 days.

Keywords: *genethic algorithm, scheduling, disability student, assistance*

1. PENDAHULUAN

Penjadwalan merupakan hal penting bagi setiap institusi karena dengan adanya penjadwalan yang baik, maka sumber daya manusia, ruang, dan waktu dapat dialokasikan untuk mendukung pekerjaan secara optimal. Begitu juga dengan Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB) membutuhkan penjadwalan damping

mahasiswa difabel yang baik. PSLD UB adalah sebuah lembaga yang berfungsi sebagai pusat penelitian tentang isu-isu disabilitas dan pemberian layanan bagi penyandang disabilitas di Universitas Brawijaya. Saat ini penjadwalan yang dilakukan oleh PSLD UB masih menggunakan cara manual. Berdasarkan data jadwal mahasiswa difabel dan semester ganjil 2015/2016, PSLD UB mempunyai 50 mahasiswa difabel dengan berbagai jenis kecacatan dan 58 *volunteer* dari berbagai

macam fakultas di Universitas Brawijaya. Dikarenakan kompleksnya data yang ada dan juga cara penentuan jadwal yang masih manual, sering muncul masalah-masalah terkait dengan penjadwalan damping seperti, jadwal bentrok antara dengan mahasiswa difabel.

Terkait dengan masalah di atas, telah banyak penelitian dilakukan untuk menyelesaikan masalah penjadwalan dengan menggunakan Algoritma Genetika (Joko et al., 2008; Teddy et al., 2009; Spyros et al., 2005). Algoritma Genetika banyak digunakan dalam penyelesaian masalah penjadwalan karena dapat mencari solusi masalah dengan kondisi satu variabel atau multivariabel. Algoritma genetika merupakan pencarian hasil yang terbaik, yang didasarkan pada perkawinan dan seleksi gen secara alami (Made et al., 2012). Algoritma Genetika mampu menemukan solusi global optimal bahkan dalam ruang pencarian yang kompleks (Samuel et al., 2009). Algoritma Genetika dan modifikasinya telah berhasil digunakan untuk memecahkan beberapa masalah optimasi kombinatorial khususnya masalah penjadwalan (Rakesh et al., 2014).

Berdasarkan pemaparan informasi diatas, maka akan diimplementasikan Algoritma Genetika dan menggunakan beberapa constraint pelanggaran pada sebuah sistem penjadwalan damping dengan judul “Optimasi Penjadwalan Damping Mahasiswa Difabel dengan Menggunakan Algoritma Genetika”. Sistem penjadwalan ini akan membantu PS LD UB menyusun jadwal pendampingan dan juga memberikan saran penjadwalan pada kondisi tertentu, sehingga mahasiswa difabel dan volunteer merasa puas terhadap penjadwalan yang ada. Hasil sistem ini adalah penjadwalan damping mahasiswa difabel dan saran penjadwalan yang harus digunakan pada kondisi tertentu.

Manfaat penelitian ini untuk memberikan pilihan kepada PS LD Universitas Brawijaya bahwa dalam menentukan jadwal damping mahasiswa difabel ada cara yang lebih efektif dan efisien, kemudian mengoptimalkan penggunaan sumber daya sehingga pada akhirnya produktivitas PS LD Universitas Brawijaya meningkat, selain itu PS LD Universitas Brawijaya dapat memberikan jadwal tanpa bentrok sehingga memudahkan para pendamping.

2. LANDASAN PERPUSTAKAAN

2.1 Penjadwalan

Penjadwalan adalah pengalokasian, pemenuhan constraints, pemberian sumber daya untuk obyek-obyek yang diletakkan dalam ruang dan waktu, sedemikian rupa untuk memenuhi tujuan yang diinginkan (Spyro et al., 2005). Penjadwalan merupakan hal umum ditempat kerja, tujuannya agar tugas dapat diselesaikan dengan cepat dengan membagi sumber daya secara tepat (Yumarsono, 2008). Penjadwalan kelas mata kuliah adalah suatu sistem penempatan waktu dan ruang dalam proses kegiatan belajar mengajar (Mariana et al., 2013). Penjadwalan di dalam universitas diartikan sebagai masalah dalam menciptakan jadwal kuliah dan dalam memenuhi constraint dengan sumber daya yang terbatas (Teddy et al., 2009). Penyusunan jadwal secara manual cenderung membutuhkan waktu yang lebih lama dan ketelitian yang cukup bagi pembuat jadwal, sehingga perlu adanya penjadwalan otomatis (Rasoul et al., 2014).

Membuat penjadwalan otomatis adalah tugas yang sangat penting karena dapat menghemat waktu kerja karyawan di suatu institusi dan perusahaan, memberikan solusi yang optimal dan cepat dalam memenuhi constraints, dapat meningkatkan produktivitas, kualitas pendidikan, kualitas pelayanan, dan juga kualitas hidup (Spyro et al., 2005). Dalam penyusunan jadwal kelas mata kuliah terdapat istilah constraints yang berarti syarat atau ketentuan. Fungsi dari constraints dalam masalah penjadwalan adalah sebagai aturan atau syarat ketentuan agar tidak terjadi bentrok dalam suatu penyusunan jadwal (Mariana et al., 2013).

Tujuan dari penjadwalan adalah bagaimana mengatur sejumlah komponen dengan tetapan dan syarat tertentu dalam satu waktu (Komang, 2010). Tujuan dari penyusunan jadwal mata kuliah adalah dihasilkannya jadwal yang efisien. Tujuan penjadwalan meningkatkan produktivitas mesin, mengurangi persediaan barang setengah jadi dengan jalan mengurangi jumlah rata-rata tugas yang menunggu dalam antrian, mengurangi keterlambatan karena batas waktu telah dilampaui (Citra, 2007). Tujuan penjadwalan adalah untuk mengoptimalkan satu atau beberapa tujuan (Dwi, 2013).

2.2 Algoritma Genetika

Menurut (Mahmudy, 2013) dalam bidang industri manufaktur, Algoritma Genetika digunakan untuk perencanaan dan penjadwalan produksi. Algoritma Genetika juga bisa diterapkan untuk kompresi citra, optimasi penugasan mengajar bagi dosen, penjadwalan dan alokasi ruang ujian, optimasi penjadwalan kuliah, optimasi multi traveling salesman problem (MT-TSP), dan penyusunan rute jadwal kunjungan wisata yang efisien.

Kelebihan Algoritma Genetika sebagai metode optimasi adalah sebagai berikut (Mahmudy, 2013):

- a) Algoritma genetika merupakan algoritma yang berbaris populasi yang memungkinkan digunakan pada optimasi masalah dengan ruang pencarian (search phase) yang sangat luas dan kompleks. Properti ini juga memungkinkan Algoritma Genetika untuk melompat keluar dari daerah optimum lokal.
- b) Individu yang ada pada populasi bisa diletakkan pada beberapa sub-populasi yang diproses pada sejumlah computer secara parallel. Hal ini bisa mengurangi waktu komputasi pada masalah yang sangat kompleks. Penggunaan sub-populasi juga bisa dilakukan pada hanya satu komputer untuk menjaga keragaman populasi dan meningkatkan kualitas hasil pencarian.
- c) Algoritma genetika menghasilkan himpunan solusi optimal yang sangat berguna pada penyelesaian masalah dengan banyak obyektif.
- d) Algoritma Genetika dapat digunakan untuk menyelesaikan masalah yang kompleks dengan banyak variabel. Variabel tersebut bisa kontinyu, diskrit atau campuran keduanya.
- e) Algoritma genetika menggunakan kromosom untuk mengkodekan solusi sehingga bisa melakukan pencarian tanpa memperhatikan informasi derivative yang spesifik dari masalah yang diselesaikan.

Algoritma Genetika dapat diimplementasikan pada sebagai macam data. Beberapa penelitian membuktikan bahwa hybrid algoritma genetika sangat efektif untuk menghasilkan solusi yang lebih baik.

2.3 Struktur Algoritma Genetika

Proses umum yang terjadi pada algoritma genetika adalah inisialisasi, reproduksi, evaluasi, dan seleksi (Mahmudy, 2013).

2.3.1 Inisialisasi

Merupakan suatu proses awal dalam algoritma genetika yaitu untuk menentukan ukuran populasi, kemudian membangkitkan individu secara acak. Setelah ukuran populasi ditentukan, kemudian tentukan nilai acak untuk setiap gen dari setiap kromosom yang ada. Ukuran populasi yang besar akan meningkatkan kemampuan untuk mengeksplorasi tiap individu yang ada dalam populasi. Ukuran populasi yang semakin kecil akan memberikan kompleksitas komputasi yang rendah karena, tidak cukup banyak individu yang ada dalam populasi. Jadi ukuran populasi awal akan menentukan kompleksitas komputasi dan eksplorasi.

2.3.2 Reproduksi

Merupakan suatu cara dalam algoritma genetika untuk menghasilkan keturunan dari tiap individu-individu yang ada di dalam pada populasi tertentu. Keturunan yang dihasilkan ini akan berevolusi dan individu yang mampu beradaptasi dengan lingkungan akan mempunyai peluang yang lebih besar untuk melewati seleksi dan akan tetap bertahan hidup. Individu yang baik akan menghasilkan keturunan yang baik juga. Reproduksi dalam algoritma genetika menggunakan operator *crossover* dan *mutation*.

2.3.3 Evaluasi

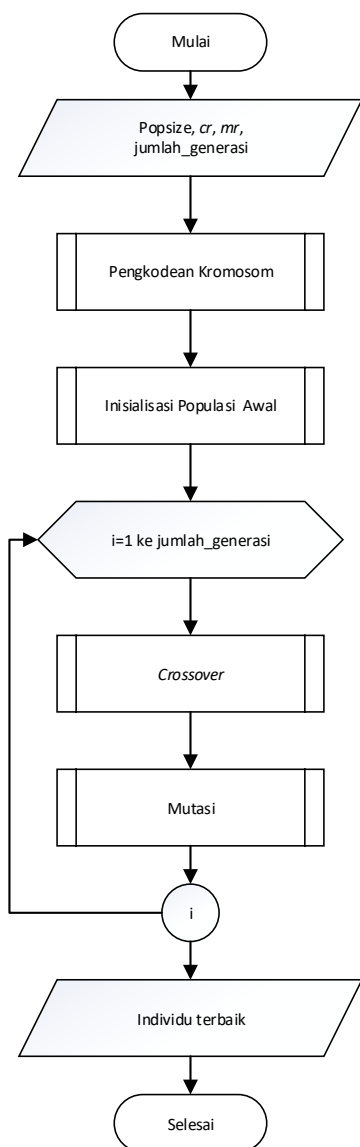
Evaluasi berfungsi untuk mendapatkan generasi terbaik berikutnya dengan memberikan suatu kondisi yang dapat memaksimalkan nilai *fitness* pada setiap kromosom. Algoritma genetika mempunyai tujuan untuk memaksimalkan nilai *fitness* atau mencari nilai *fitness* yang maksimal. Semakin individu memiliki nilai *fitness* yang tinggi maka akan dipertahankan untuk tetap hidup, sedangkan individu yang memiliki *fitness* rendah tidak akan terpilih dan akan diganti dengan individu yang memiliki nilai *fitness* lebih baik.

2.3.4 Seleksi

Pada tahap seleksi ini akan dipilih individu-individu yang baik dengan menerapkan proses persilangan dan mutasi untuk memilih individu-

individu yang terbaik dan akan terpilih. Proses seleksi bertujuan untuk mendapatkan calon induk yang baik sehingga keturunan yang dihasilkan juga akan menjadi baik. Proses seleksi ini diawali dengan mencari *fitness*. Setiap individu yang ada dalam populasi akan dihitung nilai *fitness*-nya. Nilai *fitness* akan digunakan untuk tahap seleksi berikutnya.

3. PERANCANGAN



Gambar 1. Diagram Alir Penelitian

3.1 Batasan Penelitian

1. Data yang digunakan adalah data jadwal dan mahasiswa difabel semester ganjil periode 2015/2016 yang diperoleh dari PSLD UB.
2. Jadwal volunteer yang digunakan adalah jadwal kuliah ditambah jadwal kegiatan mahasiswa.

3. Setiap volunteer bertanggung jawab mendampingi mahasiswa difabel minimal 3 kali dalam seminggu.
4. Penelitian algoritma genetika penjadwalan damping tidak memperhitungkan kelas.
5. Untuk jadwal ketersediaan pendamping dibagi menjadi 5 waktu dalam sehari.
6. Jadwal yang digunakan adalah jadwal kuliah mahasiswa dan pendamping dari hari Senin sampai hari Jumat.

Data yang digunakan dalam penelitian ini didapatkan dari PSLD Universitas Brawijaya yang meliputi:

1. Data jadwal kuliah mahasiswa difabel.
2. Data jadwal kuliah pendamping.

3.2 Implementasi Algoritma Genetika

Proses implementasi algoritma genetika untuk optimasi penjadwalan damping mahasiswa difabel adalah sebagai berikut:

1. Inisialisasi parameter awal
Parameter algoritma genetika: jumlah generasi, ukuran populasi, *Crossover rate* (*Cr*), dan *Mutation rate* (*Mr*).
2. Bangkitkan populasi awal secara random sebanyak jumlah populasi yang ditentukan.
3. Membentuk populasi baru dengan langkah-langkah sebagai berikut:
 - Melakukan proses *crossover* pada induk yang terpilih berdasarkan *cr* untuk mendapatkan anak (offspring) dengan metode one cut point.
 - Melakukan proses mutasi pada induk yang terpilih berdasarkan *mr* untuk mendapatkan anak (offspring) dengan cara memilih dua gen secara acak kemudian menukar informasi gen tersebut.
 - Menghitung nilai *fitness* untuk masing-masing kromosom atau individu.
 - Melakukan seleksi elitism untuk memilih individu sebanyak jumlah populasi awal dari gabungan individu induk dan anak untuk dijadikan populasi pada generasi selanjutnya, jika kondisi akhir terpenuhi, maka iterasi berhenti dan solusi terbaik adalah populasi yang terpilih pada generasi tersebut, jika kondisi akhir tidak terpenuhi maka lanjut ke iterasi generasi selanjutnya.

3.3 Daftar Waktu Pendampingan

Data waktu pendampingan berisi banyaknya slot waktu pendampingan yang disesuaikan dengan porsi waktu kuliah mahasiswa difabel. Maksud dari porsi waktu

disini adalah tiap satu shift pendampingan berdurasi 90 menit. Kegiatan perkuliahan mahasiswa difabel untuk hari Senin dimulai pada shift ke-1 yaitu jam 06.30-08.00 sampai shift ke-9 yaitu jam 18.30 - 20.30 dijelaskan pada Tabel 1.

Tabel 1. Rancangan Shift Pendampingan

Jam/Hari	Senin	Selasa	Rabu	Kamis	Jumat
06.30 – 08.00					
08.00 – 09.30					
;	;	;	;	;	;
18.30 – 20.30					

3.4 Alur Penyelesaian Masalah Menggunakan Algoritma Genetika

Dalam penyelesaian masalah untuk penjadwalan damping mahasiswa difabel ini digunakan alur algoritma genetika.

3.4.1 Representasi Kromosom

Representasi kromosom merupakan pengkodean kromosom digunakan dalam penjadwalan damping mahasiswa difabel. Pengkodean kromosom ini menggunakan pengkodean permutasi. Representasi dari pengkodean permutasi menggunakan bilangan bulat. Inisialisasi dari kromosom direpresentasikan ke dalam bentuk array yang berisi kode tenaga pendamping. Contoh dari inisialisasi kromosom dapat dilihat pada Gambar 2.

P1	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17	18	19	

Gambar 2. Inisialisasi Kromosom

3.4.2 Perhitungan Nilai Fitness

Nilai *fitness* didapatkan dari hasil perhitungan pelanggaran terhadap aturan yang telah ditetapkan. Penalti adalah nilai pelanggaran yang tidak sesuai dengan aturan. Nilai pelanggaran dihitung berdasarkan kemunculan pelanggaran pada kromosom. Setiap kemunculan dihitung 1 pelanggaran. Jenis pelanggaran beserta nilai pelanggaran dalam penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Nilai *Penalty* pada setiap pelanggaran

No.	Pelanggaran	Nilai
1	Seorang pendamping dalam hari dan waktu yang tidak bisa mendampingi pada waktu tersebut	50
2	Seorang pendamping berada pada kelas yang berbeda dalam hari dan waktu yang sama	20

Untuk mengetahui bobot dari masing masing individu dilakukan perhitungan nilai *fitness*. Nilai *fitness* ini menunjukkan kualitas dari masing-masing individu. Hasil perhitungan *fitness* ini kemudian digunakan untuk masukan pada proses seleksi dalam mencari individu terbaik yang akan menjadi solusi penyelesaian masalah. Untuk menghitung nilai *fitness* digunakan persamaan 1:

$$Fitness = \frac{1000}{1 + \sum \text{pelanggaran}} \quad (1)$$

Keterangan:

Pelanggaran = nilai yang ditentukan sebelumnya berdasarkan pelanggaran yang dilakukan.

Fungsi *fitness* pada Persamaan 1 merupakan persamaan yang digunakan untuk menghitung *fitness* setiap individu. Persamaan *fitness* tersebut berfungsi agar tidak ada individu yang mempunyai nilai *fitness* = 0. Karena individu yang memiliki nilai obyektif terkecil tidak akan pernah terpilih, hal ini dikarenakan nilai probabilitasnya adalah 0.

3.4.3 Crossover

Langkah-langkah metode one cut-point crossover:

1. Memilih induk secara random
2. Memasangkan kromosom yang terpilih menjadi induk untuk melakukan proses crossover.
3. Menentukan titik potong crossover secara acak.
4. Setelah ditentukan titik crossover kemudian tukar gen-gen antar 2 induk kromosom untuk menghasilkan offspring.

Ilustrasi metode one cut-point crossover dapat dilihat pada Gambar 3.

P1	3	1	3	2	1	1	2	3	1	1
P2	2	2	1	3	1	3	2	1	3	3
C	3	1	3	2	1	3	2	1	3	3

Gambar 3 Crossover one-cut point

3.4.4 Mutasi

Metode mutasi yang digunakan yaitu *reciprocal exchange mutation*. Langkah langkah dari metode mutasi adalah sebagai berikut:

1. Memilih induk secara random.
2. Memilih dua gen secara acak pada kromosom kemudian menukar nilai gen tersebut.

Ilustrasi *reciprocal exchange mutation* dapat dilihat pada Gambar 4 berikut.

P	3	1	3	2	1	1	2	3	1	1
C	3	1	1	2	1	3	2	3	1	1

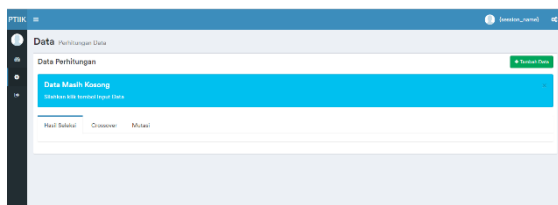
Gambar 4. *Reciprocal Exchange Mutation*

3.3.1 Seleksi

Proses seleksi dilakukan dengan menggunakan metode *elitism*. Proses seleksi dilakukan untuk menyaring semua individu hasil proses algoritma genetika untuk membentuk generasi baru. Proses seleksi menggunakan metode *elitism selection* yaitu dengan memilih kromosom dengan nilai *fitness* terbesar sebanyak *popsiz* yang telah ditentukan sebelumnya.

4. IMPLEMENTASI

4.1 Halaman Menu Utama



Gambar 5 Tampilan Antarmuka Beranda Utama

Menu tab input dipilih sebagai menu utama dimana pada menu tersebut *user* harus memasukan parameter yang dibutuhkan. Tampilan menu utama dijelaskan pada Gambar 5.

4.2 Halaman Menu Seleksi

Setelah memasukkan parameter di menu input dan menekan tombol proses, *user* akan melihat tampilan menu seleksi. Tampilan halaman menu seleksi dijelaskan pada Gambar 6.

No	Posisi	Gen	Pg.1	Pg.2	Pg.3	Perhitungan	Total Pg.	Nilai Fitness
1	c5	11132122211111111111	0	0	0	$(0 \times 90) + (0 \times 20) + (0 \times 10)$	0	9000
2	c6	11132122211111111111	0	0	0	$(0 \times 90) + (0 \times 20) + (0 \times 10)$	0	9000
3	c7	11132122211111111111	0	0	0	$(0 \times 90) + (0 \times 20) + (0 \times 10)$	0	9000
4	p1	11132122211111111111	0	0	0	$(0 \times 90) + (0 \times 20) + (0 \times 10)$	480	91,0000000000000000

Gambar 6. Tampilan Antarmuka Seleksi

Menu seleksi akan menampilkan hasil proses algoritma genetika yang sudah diurut berdasarkan dengan nilai *fitness* tertinggi.

4.3 Halaman Menu Crossover

Setelah melihat hasil seleksi, *user* dapat memilih tab menu *crossover*. Tampilan menu *crossover* dijelaskan pada Gambar 7.

Crossover p3 dan p4	
Nama Individu	Nilai
p3	11132122211111111111
p4	11132122211111111111
c1	11132122211111111111

Gambar 7 Tampilan Antarmuka Crossover

Menu *crossover* menampilkan jumlah *offspring* dari hasil proses *crossover*. Jumlahnya bergantung pada banyaknya populasi dan *Cr*.

4.4 Halaman Menu Mutasi

Mutasi c3	
Nama Individu	Nilai
c3	11132122211111111111
c3	11132122211111111111

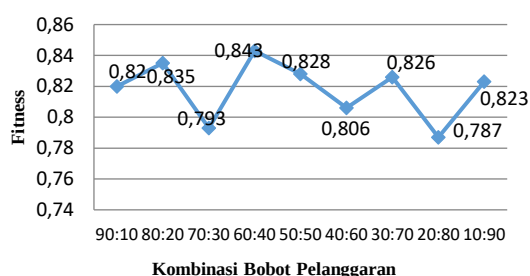
Gambar 8 Tampilan Antarmuka Mutasi

Setelah melihat *offspring* dari *crossover*, *user* dapat memilih tab menu mutasi. Tampilan menu mutasi dijelaskan pada Gambar 8. Menu mutasi menampilkan *offspring* dari proses mutasi. Jumlahnya bergantung pada banyaknya populasi dan *Mr*.

5. PENGUJIAN DAN PEMBAHASAN

5.1. Pengujian Bobot Pelanggaran

Pengujian bobot pelanggaran bertujuan untuk mengetahui perbandingan nilai *fitness* dari beberapa skenario bobot pelanggaran. Pada pengujian ini, parameter-parameter yang digunakan yaitu 4 populasi, 30 generasi, *Cr* 0.5 dan *Mr* 0.1. Pada Gambar 9 dijelaskan grafik dari pengujian bobot pelanggaran.

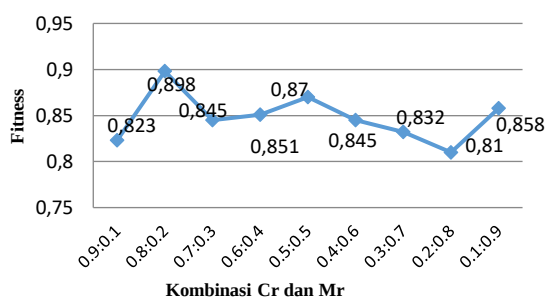


Gambar 9 Grafik Pengujian Bobot Pelanggaran

Kombinasi bobot yang digunakan pada pengujian ini adalah dengan merubah bobot pelanggaran1 dan pelanggaran2 yakni 90:10, 80:20, 70:30, 60:40, 50:50, 40:60, 30:70, 20:80, 10:90. Tiap kombinasi bobot diuji sebanyak 10 kali. Hasil dari pengujian didapatkan nilai *fitness* tertinggi mencapai 0.843.

5.2. Pengujian Kombinasi *Cr* dan *Mr*

Pengujian kombinasi *Cr* dan *Mr* dilakukan untuk mengetahui kombinasi nilai *Cr* dan *Mr* dengan nilai *fitness* terbesar. Pada pengujian ini, parameter yang digunakan yaitu 4 populasi dan 2 generasi, bobot pelanggaran pertama dan kedua 50 dan 20. Pada Gambar 10 dijelaskan grafik pengujian *Cr* dan *Mr*.



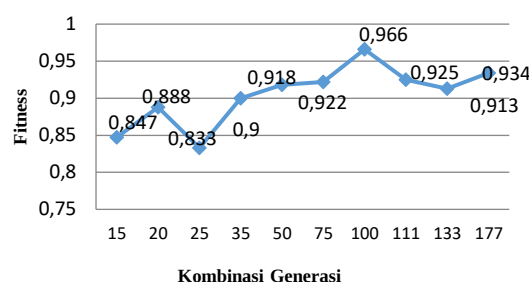
Gambar 10 Grafik Pengujian *Cr* dan *Mr*

Kombinasi *Cr* dan *Mr* yang digunakan yakni 0.9:0.1, 0.8:0.2, 0.7:0.3, 0.6:0.4, 0.5:0.5, 0.4:0.6, 0.3:0.7, 0.2:0.8, 0.1:0.9. Nilai *fitness*

rata-rata yang didapatkan sangat bervariasi. Belum ada ketetapan *Cr* dan *Mr* yang digunakan untuk memperoleh solusi optimal. Pada permasalahan kali ini, kombinasi *Cr* 0.9 dan *Mr* 0.1 menghasilkan rata-rata nilai *fitness* tertinggi yaitu 0.898.

5.3. Pengujian Generasi

Pengujian generasi dilakukan untuk mengetahui jumlah generasi dengan nilai *fitness* terbesar. Pada pengujian ini, parameter yang digunakan yaitu 30 populasi, *Cr* 0.5, *Mr* 0.2 dan generasi yang digunakan adalah 15, 20, 25, 35, 50, 75, 100, 111, 133, dan 177. Bobot pelanggaran pertama dan kedua 50 dan 20. Pada Gambar 11 dijelaskan grafik dari pengujian generasi.



Gambar 11 Grafik Pengujian Generasi

Bila diperhatikan dari grafik hasil pengujian generasi, pada generasi ke 25 nilai *fitness* sempat turun dikarenakan randomnya inisialisasi populasi, namun kemudian grafik mengalami kenaikan sampai pada generasi ke 100 mengalami peningkatan nilai *fitness* yang tinggi yaitu 0.966. Dengan demikian, banyaknya proses generasi berpengaruh terhadap nilai *fitness*. Sehingga bila diperhatikan, semakin banyak proses generasi dilakukan artinya semakin besar peluang kromosom untuk melakukan perkawinan silang untuk mendapatkan calon individu baru yang unggul.

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil implementasi dan pengujian penjadwalan mata pelajaran dengan menggunakan metode Algoritma Genetika, maka dapat diambil kesimpulan berupa:

1. Pada penelitian ini dengan penerapan Algoritma Genetika dalam penjadwalan damping mahasiswa difabel menggunakan

- pengkodean permutasi. Pengkodean ini berdasarkan kode pendamping. Panjang satu kromosom yakni 19 gen. Metode *crossover* yang digunakan adalah one-cut point. Untuk mutasi menggunakan metode reciprocal exchange *mutation*. Sedangkan untuk seleksi menggunakan metode seleksi elitism. Solusi yang didapatkan dilihat dari nilai *fitness*-nya, apabila nilai *fitness* semakin besar maka semakin baik solusi jadwal yang dihasilkan. Begitu juga sebaliknya, apabila nilai *fitness* semakin kecil maka hasil jadwal yang dihasilkan kurang baik. Ada dua pelanggaran yakni pelanggaran ketika seorang pendamping dalam hari dan waktu yang tidak bisa mendampingi pada waktu tersebut, dan ketika seorang pendamping berada pada kelas yang berbeda dalam hari dan waktu yang sama.
2. Parameter-parameter Algoritma Genetika yang ada dalam penelitian ini adalah jumlah populasi (*popsi*), *crossover rate* (*Cr*), *mutation rate* (*Mr*), dan jumlah generasi. Pengujian yang telah dilakukan pada penelitian ini adalah pengujian kombinasi bobot pelanggaran, pengujian kombinasi *Cr* dan *Mr*, dan pengujian generasi. Semua pengujian dilakukan sebanyak 10 kali. Hasil pengujian kombinasi bobot pelanggaran dengan menggunakan jumlah 30 populasi, 100 generasi, dan kombinasi *Cr* dan *Mr* 0.5 dan 0.2 menghasilkan nilai *fitness* optimal yakni 0.966. Ditemukan fakta pada penelitian ini bahwa semakin banyak jumlah generasi yang dilakukan, semakin baik nilai *fitness* yang dihasilkan. Rata-rata nilai *fitness* yang dihasilkan masih bisa lebih optimal.

6.2. Saran

Disarankan pada penelitian selanjutnya gunakan generasi seoptimal mungkin terkait dengan waktu komputasi yang optimal, dan perhatikan juga nilai *crossover rate* serta *mutation rate* nya. Karena *crossover rate* dan *mutation rate* akan mempengaruhi banyaknya individu baru yang dihasilkan dalam satu generasi. Maka sebaiknya menggunakan nilai adaptive untuk keduanya.

DAFTAR PUSTAKA

- Bartak, R., 1999. Constraint Programming: In Pursuit of The Holy Grail.
- Buliali, J.L., Herumurti, D., Wriapradja, G., 2008. Penjadwalan Matakuliah Dengan Menggunakan Algoritma Genetika dan Metode Constraint Satisfaction
- Faraji Rasoul. dan Reza Naji Hamid, 2013. An efficient *crossover* architecture for hardware parallel implementation of genetic algorithm.
- Faris, Muhammad Zaini. 2013. Optimasi Penjadwalan Mata Pelajaran Menggunakan Genetic Algorithm. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang
- Ginting, R., 2009. Penjadwalan Mesin.
- Jaya Bangun, P.B., Octarina, S. dan Ahta Virgo, G., 2012. Penerapan Konsep Algoritma Genetika untuk Penjadwalan Kegiatan Perkuliahan Semester Ganjil Kurikulum 2012 di Jurusan Matematika FMIPA UNSRI.
- Kazarlis, S. dan Petridis, V, 2004. Solving University Timetabling Problems Using Advanced Genetic Algorithms.
- Kurnia Mawaddah, N., 2006. Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika.
- Larget, H., 2012. Genetic Algorithm Used in Timetable management.
- Lukas, S., Aribowo, A, dan Muchri, M., 2012. Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan.
- Madsen, J.P., 2003. Methods for Interactive Constraint Satisfaction.
- Mahmudy, Wayan Firdaus. 2013. Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang
- Mariana, Hiryanto, L., 2008. Penjadwalan Kelas Matakuliah Menggunakan Vertex Graph Coloring dan Simulated Annealing.
- Muhyi, Y., 2008. Penjadwalan Kuliah otomatis dengan Constraint Programming.

- Pillay, N., Banzhaf, W., 2010. An Informed Genetic Algorithm for Examination Timetabling Problem.
- Rakesh, P., Gupta, D.K., Mishra, P., 2014. A New Hybrid Algorithm for University Course Timetable Problem Using Events Based On Groupings of Students.
- Sharma, Anshul, dkk. 2013. Review Paper of Various Selection Methods in Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013. ISSN: 2277 128X
- Siriwardene, N.R., 2006. Selection of genetic Algorithm Operators for Urban Drainage Model Parameter Optimisation.
- Van Peteghem, V. dan Vanhoucke, M., 2010. A Genetic Algorithm for the Preemptive and non-Preemptive Multi-Mode Resource-Constrained Project Scheduling Problem.
- Wijaya, T. dan Manurung, R., 2008. Solving University Timetabling As a Constraint Satisfaction with Genetic Algorithm.
- Yunantara, M.D., Astawa, I Gede.S., Sanjaya ER, Agus., 2012. Analisis dan Implementasi Penjadwalan Dengan Menggunakan Pengembangan Model *Crossover* dalam Algoritma Genetika.