

# Kombinasi Algoritma Genetika dengan Algoritma Palgunadi untuk Penjadwalan Mata Kuliah di Universitas Sebelas Maret

**Teno Siswono**

Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami No 36 A, Surakarta  
tennosys@student.uns.ac.id

**Sarngadi Palgunadi**

Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami No 36 A, Surakarta  
Palgunadi@uns.ac.id

**Afrizal Doewes**

Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami No 36 A, Surakarta  
afrizal.doewes@staff.uns.ac.id

## ABSTRAK

*Proses penjadwalan mata kuliah menjadi kegiatan sangat penting bagi kegiatan belajar mengajar dalam universitas, karena dengan penjadwalan baik dapat menghasilkan kegiatan belajar mengajar baik pula. Permasalahan penjadwalan mata kuliah universitas telah mendapatkan banyak perhatian dari peneliti. Terdapat beberapa metode untuk memecahkan masalah penjadwalan dan menghasilkan jadwal yang optimum, seperti algoritma genetika dan algoritma Palgunadi.*

*Pada penelitian ini diusulkan pendekatan baru dengan mengkombinasi algoritma genetika dengan algoritma Palgunadi. Pendekatan ini bertujuan untuk memperbaiki kelemahan-kelemahan algoritma genetika dan mempercepat proses algoritma genetika untuk menyelesaikan masalah penjadwalan mata kuliah di Universitas Sebelas Maret Surakarta. Studi kasus yang diambil pada penelitian ini adalah pada Jurusan Informatika dan Fisika di Universitas Sebelas Maret Surakarta, dengan data penjadwalan semester genap periode februari-juli 2013.*

*Dari hasil percobaan algoritma tersebut, disimpulkan bahwa kombinasi algoritma genetika dengan algoritma Palgunadi berhasil memperbaiki kelemahan algoritma genetika dengan secara signifikan mengurangi pelanggaran batasan dan mempersingkat waktu prosesnya.*

## Kata Kunci

Penjadwalan Mata Kuliah, Algoritma Genetika, Algoritma Palgunadi, Kombinasi

## 1. PENDAHULUAN

Jadwal merupakan tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yg terperinci (KBBI). Pihak universitas dalam menyusun penjadwalan seringkali dihadapi dengan banyaknya jumlah mata kuliah dan kelas disertai dengan terbatasnya tenaga mengajar dan ruang perkuliahan membuat proses pembentukan jadwal menjadi sangat sulit. Selain itu dalam penjadwalan terdapat berbagai jenis batasan dan aturan yang terlibat.

Pada umumnya terdapat tiga jenis teknik digunakan dalam menyelesaikan masalah penjadwalan yaitu [2]; (1) Teknik enumeratif: dalam ruang pencarian terbatas, algoritma pencarian ini menggunakan sebuah fungsi untuk melihat nilai hasil di setiap titik dan ruang satu persatu. Dengan ruang kemungkinan dari

waktu yang terbatas ini, teknik pencarian memiliki kemungkinan gagal. Contohnya adalah *dynamic programming* dan algoritma Palgunadi, (2) Teknik random: berdasarkan proses evolusi, seperti *simulated annealing* yang menggunakan proses acak untuk membantu bentuk dari pencarian energi minimum. Algoritma genetika adalah contoh prosedur pencarian yang menggunakan pilihan acak sebagai alat untuk memandu pencarian dengan ruang cari yang luas dan eksploratif melalui pengkodean dari ruang parameter. Namun teknik random ini tidak selalu menjamin untuk mencapai solusi yang optimum, (3) Metode berbasis kalkulus: dibagi menjadi dua kelas utama: langsung dan tidak langsung. Metode tidak langsung mencari ekstrem lokal dengan memecahkan set biasanya persamaan non-linear yang dihasilkan dari pengaturan gradien dari fungsi tujuan sama dengan nol seperti algoritma *Greedy*. Metode langsung mencari optima lokal dengan melompat pada fungsi dan bergerak ke arah yang berkaitan dengan gradien lokal seperti *hill-climbing*.

Algoritma genetika membentuk sebuah populasi dari set solusi permasalahan berupa kromosom-kromosom yang berkembang biak menjadi populasi baru dengan menggunakan seleksi alam dan operator genetik seperti *crossover* dan *mutation*. Sementara Algoritma Palgunadi merupakan algoritma untuk menyelesaikan permasalahan penjadwalan mata kuliah secara general.

Algoritma genetika telah diimplementasikan kedalam permasalahan penjadwalan di universitas oleh Ashish Jain, Sanjay R. Sutar, dan Kuldeep Kumar. Ashish Jain dalam penelitiannya "*Formulation of Genetic Algorithm to Generate Good Quality Course Timetable*" membuktikan bahwa algoritma genetika merupakan metode yang kuat untuk menyelesaikan masalah penjadwalan, namun pada penelitian ini batasan yang diambil jauh dari kondisi *real-world*. Sanjay R dalam penelitiannya "*University Timetabling based on Hard Constraints using Genetic Algorithm*" menyelesaikan masalah penjadwalan dengan berbasiskan batasan kaku, hasil penelitian ini mengatakan bahwa optimalisasi 100% dari sumber daya yang tersedia tidak *feasible* dalam kasus apapun. Kuldeep Kumar dalam penelitiannya "*Genetic Algorithm Approach to Automate University Timetable*" mendapatkan bahwa pada generasi ke 250 solusi jadwal yang didapatkan belum optimal dengan masih adanya pelanggaran batasan kaku dan batasan lunak yang terjadi. Sedangkan algoritma Palgunadi sampai saat ini belum diimplementasikan.

Pada penelitian ini penulis mengusulkan pendekatan baru dengan mengkombinasi algoritma genetika dengan algoritma

Palgunadi untuk memperbaiki kelemahan-kelemahan algoritma genetika dan mempercepat proses algoritma genetika untuk menyelesaikan masalah penjadwalan mata kuliah di Universitas Sebelas Maret Surakarta.

## 2. DASAR TEORI

### 2.1 Algoritma Palgunadi

Palgunadi, mengajukan algoritma baru untuk menyelesaikan *Timetable Construction Problem (TCP)*. Algoritma ini berdasarkan pada permasalahan penyusunan jadwal yang terdiri dari beberapa elemen yaitu: *teacher(T)*, *lessons(L)*, *class(C)*, *room(R)*, dan alokasi *timeslots(S)*[11].

Terdapat beberapa relasi yang ada dalam algoritma ini, relasi tersebut adalah:

- Relasi  $\alpha$ ,  $\alpha: L \times C \rightarrow T$ , mengatakan bahwa untuk setiap pelajaran dalam kelas tertentu harus memiliki guru setidaknya satu.
- Relasi  $\beta$ ,  $\beta: L \times C \rightarrow R \times S$ , mengatakan bahwa untuk setiap pelajaran di kelas harus dilakukan pada ruang yang tersedia dalam slot waktu tertentu.
- Relasi  $\gamma$ ,  $\gamma: R \times S \rightarrow \{0, 1\}$  adalah untuk memeriksa apakah ruangan di slot waktu sudah ditempati atau tidak.
- Relasi  $\delta$ ,  $\delta: T \times S \rightarrow \{0, 1\}$  adalah untuk memeriksa apakah seorang guru di slot waktu telah ditugaskan untuk mengajar.

Selain terdapat relasi, Algoritma ini membutuhkan beberapa matriks antara lain:

- Matriks TS. Melambangkan hubungan antara Teacher dan timeslot yang berbentuk boolean matriks dan dipetakan dalam bentuk 0 dan 1. Awalnya diberi nilai nol jika belum terjadwal dan diset menjadi 1 jika sudah terjadwal. Conflict terjadi apabila  $TS_1(t_1, s) = TS_2(t_2, s)$  dimana  $t_1 \neq t_2$  maka terjadi penugasan seorang guru dalam timeslot yang sama.
- Matriks CS. Melambangkan hubungan antara class dan timeslot yang berbentuk boolean matriks. Hampir sama dengan matriks TS.
- Empat set matriks penjadwalan, AS merupakan set elemen dalam  $\{TxLxC\}$  yang telah dijadwalkan, ANS set dari elemen  $\{TxLxC\}$  yang belum dijadwalkan, BS merupakan set elemen  $\{RxS\}$  yang telah dijadwalkan, dan ABS set elemen yang telah dijadwalkan secara benar.

Algoritma utama untuk menyelesaikan masalah penjadwalan ini adalah sebagai berikut:

#### Algorithm generalTCP (T, C, L, R, S)

- (1) Create the a-assignment (A), where  $A = T \times C \times \{L_r, U L_b\}$  and split A into two disjoint set  $A_r = T \times C \times L_r$  and  $A_b = T \times C \times L_b$
- (2) Create the b-assignment (B), where  $B = \{R_r, U R_b\} \times S$  and split B into two disjoint set  $B_r = R_r \times S$  and  $B_b = R_b \times S$
- (3) initialize the matrices TS and CS,  $AS = \emptyset, ANS = \emptyset, BS = \emptyset, ABS = \emptyset$
- (4) while ( There is an unexamined element in A ) do begin
  - (4.1) select a be an unexamined element  $\in A$ ;  $scheduled = false$ ;
  - (4.2) if  $a \in A_r$ , then begin {conducted in lecturing room}
    - (4.2.1) while (not(scheduled) or (there is an unexamined element in  $B_r$ )) do begin
      - (4.2.1.1) select b be an unexamined element  $\in B_r$ ,
      - (4.2.1.2) if  $TS(a,t, b,s) = 0$  and  $CS(a,c, b,s) = 0$  then begin { check the conflict}
 

$scheduled = true$ ;

{ no conflict}

$AS = AS + \{a\}$ ;  $BS = BS + \{b\}$ ;  $ABS = ABS + \{(a,b)\}$ ;

$TS(a,t, b,s) = 1$ ;  $CS(a,c, b,s) = 1$ ;  
 $A = A - \{a\}$ ,  $B = B - \{b\}$ ; end  
 else select b.next ;  
 {conflict still occur, try next room-timeslot}

end  
 (4.2.2) If not(scheduled) then begin  
 $ANS = ANS + \{a\}$ ;  $A = A - \{a\}$  end

else {conducted in lab }

(4.2.3) while (not(scheduled) or (there is an unexamined element in  $B_b$ )) do begin

(4.2.3.1) select b be an element  $\in B_b$

(4.2.3.2) if  $TS(a,t, b,s) = 0$  and  $CS(a,c, b,s) = 0$  then begin

$scheduled = true$ ;  
 $AS = AS + \{a\}$ ;  $BS = BS + \{b\}$ ;  
 $ABS = ABS + \{(a,b)\}$ ;  
 $TS(a,t, b,s) = 1$ ;  $CS(a,c, b,s) = 1$ ;  
 $A = A - \{a\}$ ,  $B = B - \{b\}$ ; end  
 else select b.next ;

end;

(4.2.4) If not(scheduled) then begin

$ANS = ANS + \{a\}$ ;  $A = A - \{a\}$  end

end;

### 2.2 Algoritma Genetika

Algoritma genetika merupakan metode heuristik adaptif yang memiliki gagasan untuk proses seleksi alam dan genetika berdasarkan penelitian Charles Darwin[8]. Algoritma genetika pertama kali ditemukan oleh John Holland dari University of Michigan pada tahun 1960.

Algoritma genetika yang dibuat Holland merupakan sebuah metode untuk memisahkan satu populasi kromosom (terdiri dari bit-bit 1 dan 0) ke populasi baru dengan menggunakan "seleksi alam" dan operator genetis seperti *crossover*, *mutation*, *inversion*. *Crossover* menukar bagian kecil dari dua kromosom, *mutation* mengganti secara acak nilai gen di beberapa lokasi pada kromosom, *inversion* membalikkan urutan beberapa gen yang berurutan dalam kromosom. Dasar teori inilah yang menjadi dasar kebanyakan program yang menggunakan algoritma genetika pada saat ini.[4]

Algoritma genetika memiliki komponen-komponen utama yang penting, yaitu:

- Teknik *Encoding*

Proses encoding adalah salah satu proses sulit dalam algoritma genetika. Hal ini disebabkan karena proses *encoding* untuk setiap permasalahan berbeda-beda karena tidak semua teknik *encoding* cocok untuk setiap permasalahan[9]. Dalam algoritma genetika, fungsi encoding berguna untuk memetakan objek variabel kedalam kode *string*.

Berdasarkan struktur encoding, dapat diklasifikasikan kedalam 1-dimensi dan 2-dimensi. *Encoding* biner, oktal, heksadesimal, permutasi, dan *value encoding* adalah jenis 1-dimensi, sementara encoding pohon merupakan jenis 2-dimensi[10].

- Prosedur inialisasi

Membangkitkan populasi awal adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. 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 kromosom dilakukan secara acak, namun

demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

- Proses evaluasi nilai fitness  
Evaluasi fitness merupakan dasar untuk proses seleksi. Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati.
- Proses seleksi  
Pemilihan dua buah kromosom yang dijadikan induk atau sebagai orang tua dilakukan secara proporsional sesuai dengan dengan nilai fitness-nya. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung dari nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai fitness inilah yang nantinya akan digunakan pada tahap seleksi berikutnya.
- Proses Rekombinasi atau penyilangan (*Crossover*)  
Crossover melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian untuk siap diuji. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* ( $P_c$ ) yang ditentukan secara acak dalam rentang (0,1).
- Proses Mutasi  
Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Proses mutasi dilakukan pada setiap atau sebagian individu dengan probabilitas mutasi ( $P_m$ ) yang ditentukan secara acak dalam rentang (0,1)

### 3. METODOLOGI

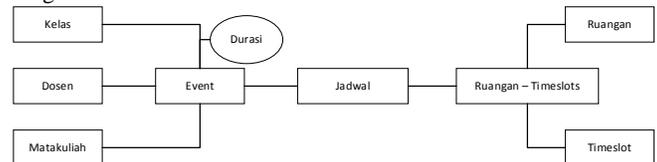
#### 3.1 Pemodelan dan Perancangan

##### 3.1.1 Pemodelan Data

Memodelkan data yang telah didapatkan menjadi model data yang terstruktur agar lebih mudah dipahami serta mengurangi data-data yang tidak perlu untuk digunakan. Skema objek penjadwalan ini secara garis besar dapat didefinisikan sebagai berikut:

- Dosen, merupakan set dari dosen  $D = \{d_1, d_2, \dots, d_n\}$
- Kelas, merupakan set dari kelas  $K = \{k_1, k_2, \dots, k_n\}$ . Kelas terdiri dari sejumlah mahasiswa. Telah ditentukan bahwa mahasiswa telah dikelompokkan dalam kelas, sehingga kelas bersifat disjoint berarti tidak ada mahasiswa yang sama dalam beberapa kelas.
- Mata kuliah, merupakan set dari mata kuliah  $M = \{m_1, m_2, m_3, \dots, m_n\}$ .
- Timeslots, merupakan set dari timeslots  $T = \{t_1, t_2, \dots, t_n\}$ . Timeslots merupakan interval waktu dimana perkuliahan akan dilakukan. Timeslots memiliki waktu mulai dan waktu akhir. Elemen dari set timeslots memiliki bentuk <hari><jam> seperti *senin1, senin2, ..., jumat10*.
- Ruang, merupakan set dari ruangan  $R = \{r_1, r_2, \dots, r_n\}$ .
- Kegiatan Perkuliahan, merupakan set dari kegiatan perkuliahan (event)  $E = \{e_1, e_2, \dots, e_n\}$ . Kegiatan perkuliahan memiliki atribut:
  - Matakuliah yang diajarkan,  $M_E = \{m\}$
  - Dosen yang mengajar di matakuliah, set dari  $D_E = \{d_1, d_2, \dots, d_n\}$
  - Kelas yang diajar di matakuliah, set dari  $K_E = \{k_1, k_2, \dots, k_n\}$

- Total durasi timeslot dalam sebuah perkuliahan,  $Dur_E = \{dur\}$
- Ruang – Timeslots, set dari ruangan-timeslots dengan alasan untuk mempersimpel proses pencarian.  $RxT = \{rxt_1, rxt_2, \dots, rxt_n\}$ , memiliki atribut
  - Ruang,  $R_{RxT} = \{r\}$
  - Waktu timeslot,  $T_{RxT} = \{t\}$
- Jadwal, set dari jadwal, Merupakan gabungan dari ruangan – timeslot dengan event.  $J = \{r_1t_1e_1, r_1t_2e_1, r_2t_1e_2, \dots, r_nt_1e_n\}$   
Secara garis besar model penjadwalan ini dapat digambarkan seperti gambar 1.



Gambar 1. Skema Sistem Penjadwalan

##### 3.1.2 Batasan-batasan Penjadwalan

Penjadwalan matakuliah di universitas ini memiliki batasan-batasan. Batasan-batasan penjadwalan dapat diklasifikasikan menjadi dua kategori, yaitu batasan kaku (*hard constraints*), dan batasan lunak (*soft constraints*). Batasan kaku merupakan batasan yang wajib dipenuhi sepenuhnya dalam penjadwalan. Sedangkan batasan lunak merupakan batasan yang dapat dilanggar tetapi dijaga agar nilai pinalty dari batasan lunak ini seminimum mungkin.

Batasan kaku terdiri atas:

- Dalam satu timeslots dan ruangan hanya boleh dijadwalkan satu perkuliahan
- Perkuliahan praktek harus dijadwalkan di kelas praktikum, dan perkuliahan teori di kelas teori
- Dosen hanya dapat mengajar satu perkuliahan dalam satu timeslots
- Kelas hanya dapat mengikuti satu perkuliahan dalam satu timeslots
- Mata kuliah tidak boleh dijadwalkan melewati jam sholat jum'at

Batasan lunak terdiri atas:

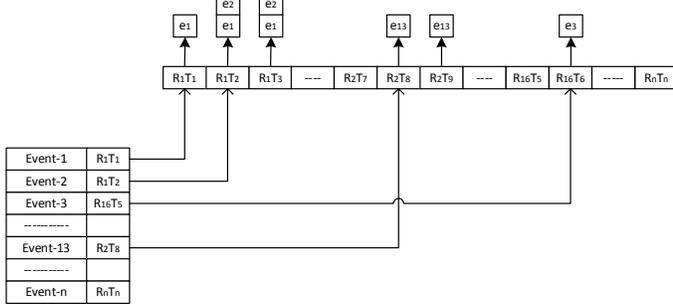
- Mata kuliah wajib pada tahun angkatan atas tidak boleh dalam satu timeslots mata kuliah praktek tahun angkatan bawahnya, dengan alasan agar mendapatkan asisten praktikum
- Mata kuliah tidak boleh dijadwalkan melewati jam istirahat
- Mata kuliah dijadwalkan pada ruangan jurusannya

### 3.2 Kombinasi Algoritma Genetika dengan Algoritma Palgunadi

Pada perancangan dan pengimplementasian ini, algoritma genetika dikombinasikan dengan algoritma Palgunadi. Tujuan utama dari proses kombinasi ini adalah untuk membuat algoritma genetika menjadi terarah proses heuristiknya. Algoritma Palgunadi berperan dalam proses penginisialisasian kromosom dan proses repair. Perbedaan algoritma genetika kombinasi ini dengan algoritma genetika biasa yaitu pada tahap inialisasi, *crossover*, dan mutasi.

#### 3.2.1 Pengkodean kromosom

Mengacu kepada skema yang telah dibuat sebelumnya, pada algoritma genetika ini kromosom direpresentasikan dengan menggunakan encoding nilai. Terdapat sebuah vektor berupa list dari setiap Event yang memiliki nilai id posisi Ruangan-Tiemeslots jam pertamanya. Selain itu terdapat sebuah vektor berupa list dari setiap Ruangan-Timeslots yang memiliki nilai berupa list id Event yang dialokasikan di Ruangan-Timeslots tersebut. Representasi kromosom dapat dilihat pada Gambar 2.



Gambar 2. Representasi Kromosom

3.2.2 Inisialisasi Kromosom

Inisialisasi kromosom pada tahap ini dilakukan dengan menggunakan prinsip algoritma Palgunadi namun constraint yang dicek hanya berupa batasan kaku. Untuk setiap Event diberikan sebuah nilai yang berupa kode RxT, lalu pada struktur tambahan setiap RxT sejumlah n durasi dari Event ditempatkan Eventnya

3.2.3 Penentuan fungsi evaluasi

Setelah sejumlah populasi telah dibentuk, setiap populasi akan dievaluasi untuk mengetahui nilai fitnessnya. Evaluasi ditentukan berdasarkan batasan kaku (*hard constraints*) dan batasan lunak (*soft constraints*) yang telah dijelaskan sebelumnya. Setiap batasan kaku memiliki nilai poin 2 dan setiap batasan lunak memiliki nilai poin 1, sehingga untuk evaluasi fitness dapat dirumuskan sebagai:

$$fitness = \frac{\sum Bk + \sum Bl}{Bkmax + Blmax}$$

Keterangan:

- Bk = Nilai batasan kaku
- Bl = Nilai batasan lunak
- Bkmax = Nilai batasan kaku maksimal
- Blmax = Nilai batasan lunak maksimal

3.2.4 Seleksi

Seleksi dilakukan untuk memilih beberapa pasang kromosom yang dijadikan induk atau sebagai orang tua untuk sejumlah n anak berikutnya yang akan menggantikan individu dalam populasi pada setiap generasi. Pemilihan pasangan kromosom dilakukan dengan menggunakan seleksi *tournament*. Pada metode seleksi dengan turnamen ini akan ditetapkan dua buah individu yang dipilih secara acak (*random*) dari suatu populasi. Individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk pertama, demikian juga untuk pemilihan induk kedua.

3.2.5 Crossover

Setelah dua kromosom induk selesai dipilih, langkah berikutnya adalah melakukan rekombinasi yaitu penyilangan (*crossover*) terhadap pasangan kromosom. Penyilangan akan menukar informasi genetik antara dua kromosom induk yang terpilih dari proses seleksi untuk membentuk sebuah kromosom anak. Proses *crossover* dilakukan dengan sebuah probabilitas *crossover* (Pc) yaitu ditentukan sebesar 0,7 (70%).

*Crossover* dilakukan dengan menggunakan *crossover* satu titik. Dimana dicari titik potong sebuah kromosom, potongan kromosom pertama berasal dari induk 1, dan sisa kromosom diambil dari induk 2 dengan dilakukan sebuah fungsi untuk mengecek pelanggaran batasan kaku. Jika terdapat pelanggaran batasan kaku, maka dilakukan perbaikan bertahap. Perbaikan ini terdiri dari dua tahap yaitu *repair 1*(R1), dan *repair 2*(R2). R1 mengganti nilai gen yang melanggar batasan kaku dengan nilai baru yang tidak melanggar. R2 menukar nilai gen yang melanggar batasan kaku dengan nilai pada gen lain sehingga tidak melanggar. Jika sebuah kromosom induk 2 melanggar hard constraint maka tahap selanjutnya yang dilakukan adalah R1 jika R1 tidak dapat memperbaiki kromosom, maka dilakukan tahap R2. Jika kedua tahap R1 dan R2 tidak bisa memperbaiki kromosom, maka kromosom anak dibuang.

3.2.6 Mutasi

Anak hasil *crossover* sebelum dilepaskan ke populasi memiliki kemungkinan untuk terjadinya mutasi. Pada proses mutasi ini pergantian nilai pada setiap gen dilakukan dengan sebuah proses pengecekan pelanggaran batasan kaku jika melanggar maka akan dicari nilai baru yang tidak melanggar hard constraint, proses ini dilakukan hingga didapat kromosom yang tidak melanggar batasan kaku

3.2.7 Pergantian kromosom

Pada tahap ini kromosom anak hasil *crossover* dan mutasi akan menggantikan posisi kromosom yang lama untuk membentuk sebuah populasi baru dengan ukuran yang sama. Pada prosedur pergantian ini diterapkan konsep *elitism* yang memastikan kromosom dengan *fitness* tertinggi tidak tersingkirkan dalam populasi.

3.3 Pengujian

Pada tahap ini dilakukan pengujian terhadap kombinasi algoritma genetika dengan algoritma Palgunadi. Sebagai perbandingan dari algoritma yang diusulkan tersebut, dilakukan pula pengujian terhadap algoritma Palgunadi dan algoritma genetika biasa. Pengujian dilakukan dengan menggunakan data jadwal jurusan informatika dan fisika semester genap periode february-juli 2013 Universitas Sebelas Maret dengan total 9 kelas mahasiswa, 59 mata kuliah, 107 perkuliahan, 17 ruangan, 5 hari kuliah, dan 10 jam kuliah.

3.3.1 Pengujian Algoritma Palgunadi

Pengujian dilakukan untuk menguji kemampuan algoritma Palgunadi dalam menyelesaikan masalah penjadwalan universitas. Setelah menjalankan algoritma Palgunadi didapatkan bahwa jadwal telah berhasil dijadwalkan dengan tidak ada pelanggaran batasan kaku dan terdapat satu pelanggaran batasan lunak dengan rata-rata waktu proses algoritma Palgunadi ini adalah 2 menit 20 detik. Dari hasil yang diperoleh, dapat disimpulkan bahwa algoritma Palgunadi bekerja untuk penjadwalan mata kuliah universitas sebagaimana seharusnya.

3.3.2 Pengujian Algoritma Genetika Biasa

Pada pengujian ini dilakukan pengujian kemampuan algoritma genetika dalam menyelesaikan masalah penjadwalan mata kuliah universitas. Setting parameter algoritma genetika yang digunakan adalah sebagai berikut:

- Populasi maksimum: 10; 20
- Generasi yang diganti: 5; 10
- Kromosom elitism: 5
- Kemungkinan terjadi crossover: 0,7
- Kemungkinan terjadi mutasi: 0,3
- Titik mutasi: 5 titik

Dari hasil berjalannya algoritma didapatkan bahwa dengan jumlah data yang banyak ternyata menghasilkan hasil yang kurang memuaskan. Selain itu waktu proses algoritma genetika sangat lama yaitu untuk 10 populasi membutuhkan waktu 3 jam 24 menit 19 detik, dan untuk 20 populasi membutuhkan waktu 7 jam 17 menit 23 detik. Lamanya waktu proses ini disebabkan karena lamanya pengecekan atau proses evaluasi dari kromosom.

**3.3.3 Pengujian Algoritma Genetika Hibridasi**

Pada pengujian ini dilakukan pengujian kemampuan hibridasi algoritma Palgunadi dan algoritma genetika dalam menyelesaikan masalah penjadwalan mata kuliah universitas. Setting parameter hibridasi algoritma genetika yang digunakan adalah sebagai berikut:

- Populasi maksimum: 10; 20
- Generasi yang diganti: 5; 10
- Kromosom elitism: 5
- Kemungkinan terjadi crossover: 0,7
- Kemungkinan terjadi mutasi: 0,3
- Titik mutasi: 5 titik

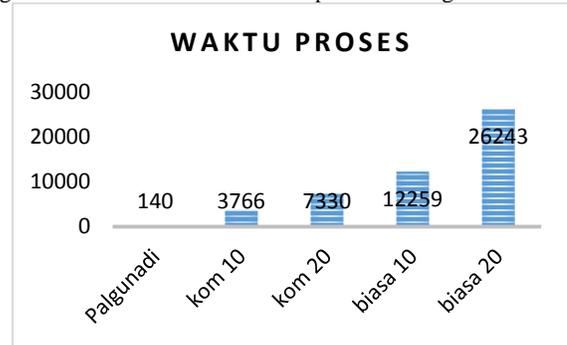
Dari hasil berjalannya algoritma didapatkan bahwa dengan menggabungkan algoritma genetika dan algoritma Palgunadi. Pada hasil algoritma ini didapatkan bahwa hasil penjadwalan memuaskan dengan jumlah pelanggaran yang jauh lebih sedikit dibanding dengan algoritma genetika biasa. Selain itu dengan mengeleminasi hard constraint, secara tidak langsung mengurangi proses dan waktu dari proses evaluasi dari kromosom, sehingga waktu proses yang ditempuh algoritma hibridasi ini adalah untuk populasi 10 membutuhkan waktu 1 jam 2 menit 46 detik, dan untuk populasi 20 membutuhkan waktu 2 jam 2 menit 10 detik.

**4. PEMBAHASAN**

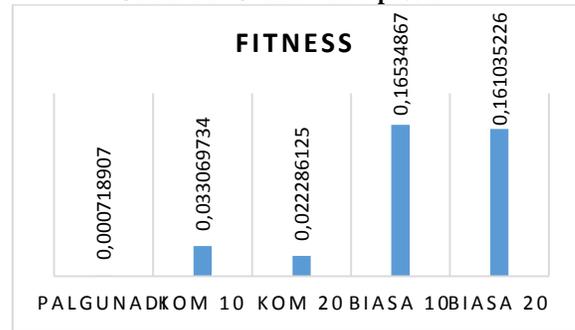
Pada percobaan ketiga algoritma tersebut dengan menggunakan dataset penjadwalan jurusan informatika dan fisika pada semester genap periode februari-juli 2013 didapatkan hasil yaitu:

Algoritma Palgunadi berhasil menjadwalkan mata kuliah dengan waktu paling singkat, dibanding dengan kedua algoritma

lainnya. Selain itu algoritma Palgunadi juga menjadwalkan dengan fitness yang lebih baik. Grafik waktu proses dapat dilihat di gambar 3. Grafik fitness akhir dapat dilihat di gambar 4.



Gambar 3. Grafik waktu proses

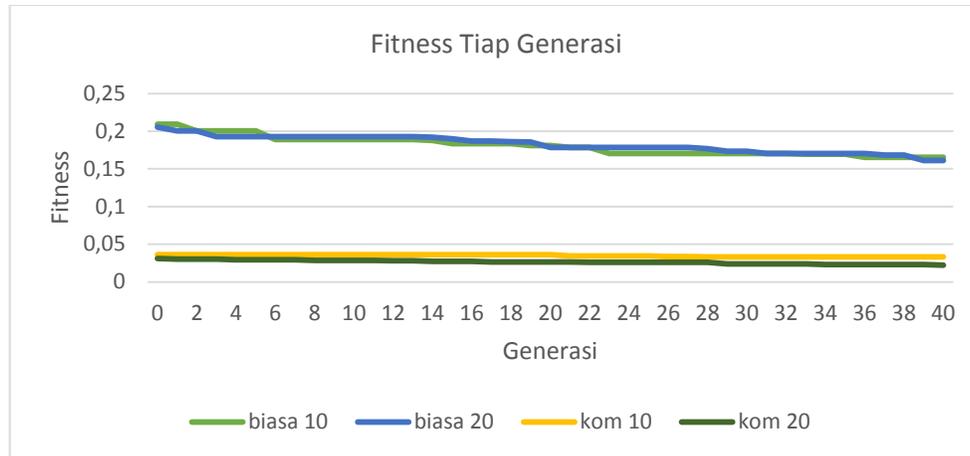


Gambar 4. Grafik fitness akhir

Untuk perbandingan algoritma genetika biasa dankombinasi algoritma genetika dengan algoritma palgunadi, didapatkan waktu proses algoritma kombinasi jauh lebih sedikit dibanding dengan algoritma genetika biasa. Selain itu nilai fitness yang didapatkan pengkombinasian ini sangat jauh lebih baik dibanding algoritma genetika biasa seperti pada tabel 1. Gambar5 menunjukkan grafik nilai fitness untuk setiap generasi pada algoritma genetika biasa dan algoritma genetika yangdikombinasi.

**Tabel 1. Nilai fitness terbaik pada setiap generasi untuk GA kombinasi gan GA biasa**

pop	Algoritma Generika Kombinasi						Algoritma genetka biasa					
	10			20			10			20		
gen	Fitness	PB.Kaku	PB.Lunak	Fitness	PB.Kaku	PB.Lunak	Fitness	PB. Kaku	PB.Lunak	Fitness	PB.Kaku	PB.Lunak
0	0.035945	0	50	0.030913	0	43	0.209202	112	67	0.205607	111	64
10	0.035945	0	50	0.028756	0	40	0.189073	99	65	0.192667	105	58
20	0.035945	0	50	0.0266	0	37	0.181165	95	62	0.178289	96	56
30	0.03307	0	46	0.023724	0	33	0.170381	89	59	0.173257	92	57
40	0.03307	0	46	0.022286	0	31	0.165349	86	58	0.161035	84	56



Gambar 5. Grafik fitness tiap generasi pada GA kombinasi dan GA biasa

## 5. KESIMPULAN

Penelitian ini telah menghasilkan sebuah kombinasi algoritma genetika dengan algoritma Palgunadi. Kombinasi ini berhasil memperbaiki proses heuristik dari algoritma genetika dibuktikan dengan semakin baiknya nilai fitness yang dihasilkan dan semakin sedikit waktu proses yang dibutuhkan. Namun pada generasi ke 40 dari kombinasi algoritma ini masih menyisakan sejumlah pelanggaran batasan lunak. Selain itu waktu proses yang dibutuhkan masih tergolong lama dibanding algoritma Palgunadi. Beberapa penelitian selanjutnya yang mungkin adalah: menambah batasan-batasan dalam penjadwalan mata kuliah, memperluas sampel data, dan melakukan kombinasi dengan algoritma lainnya agar algoritma genetika berhasil menjadwalkan dengan nol pelanggaran dan lebih mempersingkat waktu proses.

## 6. DAFTAR PUSTAKA

- [1] Achmad, Basuki. 2003. Strategi Menggunakan Algoritma Genetika. *Politeknik Elektronika Negeri Surabaya PENS-ITS. Surabaya.*
- [2] Arous, N., Ellouze, N., dan Abdallah, S. 1999. Evolutionary potential timetables optimization by means of genetic and greedy algorithms. *Proceedings of the 1999 International Conference on Information Intelligence and Systems (ICIIS)*
- [3] Davis. 1991. Handbook of Genetic Algorithms. *New York: Van Nostrand Reinhold.*
- [4] Fariza, A., Martiana, E., Sucipto H. (2006). Aplikasi Algoritma Genetika Multi Obyektif pada Traveling Salesman Problem. *Politeknik Elektronika Negeri Surabaya ITS. Surabaya.*
- [5] Fitri. 2002. Penjadwalan Perkuliahan Dengan Pengujian Tabel Waktu (Time- Tabel) Menggunakan Algoritma Genetika. *Jurnal Jurusan Teknik Informatika. Universitas Komputer Indonesia. Bandung.*
- [6] Goldberg, D. 1989. Genetic algorithm in Search, Optimization and Machine Learning. *Addison -Wesley. Reading, MA.*
- [7] Kumar, Anit. 2013. Encoding Schemes in Genetic Algorithm. *International Journal of Advanced Research in IT and Engineering. Vol. 2, No. 3, Maret 2013*
- [8] Kusumadewi, S, dan Purnoma, Hari. 2005. Penyelesaian Masalah Optimasi Menggunakan Teknik-teknik Heuristik Edisi Pertama. *Penerbit Graha Ilmu. Yogyakarta.*
- [9] Lukas, Samuel. 2005. Penerapan Algoritma Genetika Untuk Traveling Salesman Problem Dengan Menggunakan Metode Order Crossover Dan Insertion Mutation. *Seminar Nasional Aplikasi Teknologi Informasi 2005. Yogyakarta.*
- [10] Mitsuo Gen. 1996. Genetic algorithms and engineering design. *Wiley-IEEE. ISBN 0471127418.*
- [11] Palgunadi. 2012. Timetabling Construction Problem (TCP). *Jurnal ITSMART Vol.1 No.1 Juni 2012.*
- [12] Puspaningrum, W.A., Djunaidy, A., dan Vinarti, R.A. 2013. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. *Jurnal Teknik POMITS Vol. 2, No. 1, (2013) ISSN: 2337-3539.*
- [13] Sabarudin, La Ode. 2012. Algoritma Genetika (Menggunakan Matlab)
- [14] Syamsuddin, Aries. 2004. Pengenalan Algoritma Genetik, *Kuliah Umum IlmuKomputer.Com*
- [15] Siahaan, Darmansyah. 2012. Penggunaan Algoritma Genetika dalam Menentukan Penjadwalan Perkuliahan. *Jurusan Matematika. Universitas Negeri Medan. Medan.*
- [16] Thiang, Kurniawan, R., dan Ferdinando, H. Implementasi Algoritma Genetika pada Mikrokontroler MCS51 Untuk Mencari Rute Terpendek. *Jurusan Teknik Elektro. Universitas Kristen Petra. Surabaya.*