OPTIMASI PENJADWALAN CERDAS MENGGUNAKAN ALGORITMA MEMETIKA

Muhammad Arief Nugroho[1], Galih Hermawan, S.Kom., M.T.[2]

[1, 2]Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-116, Bandung 40132
E-mail : ariefn82@Gmail.com[1], galih.hermawan@gmail.com[2]

ABSTRAK

Sistem penjadwalan kuliah memiliki permasalahan yang kompleks karena adanya berbagai komponen yang terdiri dari mata kuliah, dosen, ruang kelas, dan *slot* waktu perkuliahan dengan memperhatikan batasan dan syarat tertentu yang harus dipenuhi. Pada implementasinya, ketika banyaknya dosen mata kuliah yang harus dijadwalkan maka proses penjadwalan membutuhkan waktu yang lama dan tingkat ketelitian yang tinggi. Untuk itu algoritma *memetika* diterapkan untuk menyelesaikan permasalahan ini. Menurut beberapa jurnal algoritma *memetika* dinilai lebih baik dari algoritma *heuristic* lainnya dalam menyelesaikan permasalahan penjadwalan yang besar datanya.

Adapun tahapan algoritma *memetika* yang dianalisis dimulai dari pembangunan kromosom-kromosom yang nantinya akan menjadi calon solusi lalu pencarian nilai *fitness*, seleksi, *crossover* (pindah silang), mutasi, dan pencarian lokal. Berdasarkan hasil pengujian, algoritma *memetika* mampu menyelesaikan permasalahan penjadwalan data yang banyak. Proses pencarian lokal sangat membantu dalam peningkatan kualitas dari individu tiap generasinya. Proses pencarian lokal juga dapat mengatasi nilai lokal optimal yang sering terjadi di algoritma genetika ketika data masukannya sangat banyak. Nilai parameter yang didapatkan adalah 0.95 untuk probabilitas *crossover* dan 0.10 untuk probabilitas mutasi. Didapatkan juga, ketika jumlah dosen yang memiliki *soft constraint* semakin banyak maka *soft constraint* yang dilanggar akan semakin banyak.

Kata kunci : *Memetika*, Genetika, Pencarian Lokal, *Heuristic*, Sistem Penjadwalan.

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Algoritma *Memetika* (AM) adalah perluasan dari Algoritma Genetik. Kelebihan dari AG adalah pada cara kerjanya yang paralel. AG bekerja dalam ruang pencarian yang menggunakan banyak individu sekaligus, sehingga kemungkinan AG untuk terjebak pada ekstrem lokal lebih kecil dibandingkan metode lain[6]. Metode *heuristic* merupakan satu metode penyelesaian yang menggunakan konsep pendekatan. Beberapa algoritma *heuristic* telah dikembangkan antara lain algoritma *memetika*. Algoritma *memetika* memiliki proses *local search* yang mampu menyelesaikan permasalahan penjadwalan. Menurut Albar Ali M [1] pada kasus yang memiliki *time slot* yang terbilang besar algoritma *memetika* dengan baik menyelesaikan masalah tersebut. Menurut Albar Ali M, Sadaf N. Jat, dan Shengxiang Yang [1] [2] proses pencarian lokal yang dimiliki oleh algoritma *memetika* juga sangat membantu dalam pencarian solusi optimal terdekat dengan meningkatkan kualitas dari individu. Menurut Garg Poonam [3] algoritma *memetika* lebih baik daripada algoritma genetika dalam hal pencarian ruang solusinya, menurut Ray Pillarrichie dan Suyanto [4] terdapat perbedaan dalam hal kualitas kromosom yang dihasilkan dilihat dari nilai *fitness* dari tiap generasi. Masalah penjadwalan kuliah merupakan permasalahan yang kompleks karena adanya berbagai komponen yang terdiri dari mata kuliah, dosen, ruang kelas, dan *slot* waktu perkuliahan dengan memperhatikan batasan dan syarat tertentu yang harus dijaga atau dipenuhi. Pada implementasinya, ketika banyaknya dosen mata kuliah yang harus dijadwalkan maka proses penjadwalan membutuhkan waktu yang lama dan tingkat ketelitian yang tinggi.

Untuk itu algoritma *memetika* diterapkan untuk menyelesaikan permasalahan ini. Tahapan yang ada pada algoritma *memetika* dasarnya sama dengan tahapan pada algoritma genetik, hanya saja di tambah tahapan *local search* yang kegunaannya telah di jelaskan sebelumnya. Di mana tahapannya dimulai dari pembangunan kromosom-kromosom yang nantinya akan menjadi calon solusi lalu pencarian nilai *fitness*, proses seleksi, proses *crossover* (pindah silang), dan proses mutasi lalu tahapan *local search* yang dimaksudkan untuk melakukan perbaikan lokal.

1.2 Tujuan Penelitian

Dari permasalahan yang sudah dijelaskan maka tujuan dari penelitian ini ialah Mengetahui optimasi dari algoritma *memetika* dalam menyelesaikan permasalahan pada kasus penjadwalan.

1.3 Batasan Penelitian

Studi kasus di Prodi Teknik Informatika UNIKOM Bandung. Dengan data set yang diambil ialah data semester genap tahun ajaran 2015/2016 sebanyak 325 data (tidak termasuk mata kuliah kewirausahaan).

1.4 Algoritma Genetika

Menurut David E. Goldberg [7] algoritma adalah algoritma komputasi yang di inspirasi teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi aturan "yang kuat adalah yang menang". Darwin juga menyatakan bahwa kelangsungan hidup suatu makhluk dapat dipertahankan melalui proses reproduksi, *crossover*, dan mutasi. Konsep dalam teori evolusi Darwin tersebut kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi satu permasalahan dengan cara yang lebih "alamiah". Sebuah solusi yang dibangkitkan dalam algoritma genetika disebut sebagai kromosom, sedangkan kumpulan kromosom-kromosom tersebut disebut sebagai populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi. Dalam tiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (*fungsi_objektif*) menggunakan ukuran yang disebut dengan *fitness*.

Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan seleksi. Proses seleksi kromosom menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai *fitness* tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya. Kromosom-kromosom baru yang disebut dengan *offspring*, dibentuk dengan cara melakukan perkawinan antar kromosom-kromosom dalam satu generasi yang disebut sebagai proses *crossover*. Jumlah kromosom dalam populasi yang mengalami *crossover* ditentukan oleh *paramater* yang disebut dengan *crossover_rate* (probabilitas *crossover*). Mekanisme perubahan susunan unsur penyusun makhluk hidup akibat adanya faktor alam yang disebut dengan mutasi direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam kromosom dengan suatu nilai acak. Jumlah gen dalam populasi yang mengalami mutasi ditentukan oleh parameter yang dinamakan *mutation_rate* (probabilitas mutasi). Setelah beberapa generasi akan dihasilkan kromosom-kromosom yang nilai gen-gennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh algoritma genetika terhadap permasalahan yang ingin diselesaikan.

1.5 Algoritma *Memetika*

Algoritma *Memetika* (AM) adalah perluasan dari Algoritma Genetika. Kelebihan dari AG adalah pada cara kerjanya yang paralel. AG bekerja dalam ruang pencarian yang menggunakan banyak individu sekaligus, sehingga kemungkinan AG untuk terjebak pada ekstrem lokal lebih kecil dibandingkan metode lain. Kekurangan dari AG adalah dalam hal waktu komputasi karena harus melakukan evaluasi *fitness* pada semua solusi di setiap iterasi, sehingga AG bisa lebih lambat dibandingkan dengan metode lain. [5] Namun telah diketahui bahwa AG tidak cukup baik untuk mencari solusi yang sangat dekat dengan solusi optimal. Kekurangan ini dapat di akomodasi dengan cara menambahkan metode *Local Search* pada AG. Hasil penggabungan ini dikenal dengan sebutan Algoritma *Memetika* (AM). AM adalah suatu metode pencarian *heuristic* yang memiliki karakteristik yang sama dengan AG dikombinasikan dengan metode *Local Search* yang secara bersama-sama dapat meningkatkan kualitas pencarian solusi (Moscato, 1989). Pada AM, *Local Search* bertujuan untuk melakukan perbaikan lokal yang dapat diterapkan sebelum dan atau sesudah proses seleksi, *crossover*, dan mutasi. *Local Search* juga dapat berguna untuk mengontrol besarnya ruang pencarian solusi. AM dapat memberikan hasil yang lebih baik daripada AG, namun memerlukan waktu komputasi yang lebih lama[1].

1.6 Pencarian Lokal

Pencarian lokal merupakan pertukaran atau penempatan kembali informasi yang di bawa oleh gen-gen dalam satu kromosom anak dengan harapan dapat meningkatkan kualitas kromosom tersebut. Pencarian lokal dapat dilakukan dengan menukar dua gen atau permutasi beberapa gen tanpa mengurangi kualitas dari kromosom sebelumnya. Dalam kasus tertentu, pencarian lokal hanya dilakukan kepada gen yang terkena penalti, artinya hanya informasi yang kurang tepat yang dipertukarkan antar gen dalam satu kromosom yang sama[1].

Menurut Rossi-Doria Olivia dan Paetcher Ben [6], dalam penjadwalan pencarian lokal yang efektif terdiri atas dua tahap yaitu tahap memperbaiki dan tahap meningkatkan kualitas jadwal. Penjelasan tahap tersebut adalah sebagai berikut.

a. Tahap memperbaiki jadwal dari yang tidak layak menjadi layak adalah dengan mengurangi *slot* waktu yang digunakan.
b. Tahap meningkatkan kualitas jadwal layak adalah dengan mengurangi pelanggaran pada kendala *soft*.

Menurut Sadaf N. Jat, Shengxiang Yang [2], proses pencarian lokal dibagi menjadi 3 tahapan:

a. Mengganti *slot* waktu di dalam gen secara berurutan.
b. Menukar *slot* waktu dari 2 kromosom di dalam 1 individu.
c. Melakukan permutasi.

1.7 Pembangkitan

Pembentukan populasi awal yang bekerja pada populasi, yaitu kumpulan individu yang di dalamnya terdapat kromosom atau solusi yang akan diperbaharui pada setiap generasinya. Pembentukan populasi awal yang berisi kumpulan individu sebanyak ukuran populasi. Proses *inisialisasi* ini biasanya dilakukan secara acak.

1.8 Fungsi *Fitness*

Fungsi penalti Fungsi penalti memiliki nilai suatu kromosom yang menentukan kualitas atau kelayakan solusi masalah. Nilai penalti menunjukkan bobot pelanggaran terhadap suatu kendala atau batasan. Dalam masalah optimasi, nilai penalti biasanya diminimumkan sehingga diperoleh solusi dengan jumlah pelanggaran kendala yang paling sedikit. Semakin kecil nilai penalti maka akan semakin meningkatkan kualitas solusi, sedangkan kelayakan solusi ditentukan oleh kendala suatu nilai penalti. Rumus fungsi *fitness* yang digunakan adalah persamaan 1 :

$$Fitness = \frac{1}{(1+\sum_{i=1}^{5} penalty - i)} \quad (1)$$

1.9 Seleksi

Seleksi orang tua Seleksi orang tua merupakan bagian evolusi yaitu proses mendekati kandidat solusi permasalahan yang diharapkan. Seleksi orang tua adalah operator algoritma *memetika* yang dilakukan pada setiap generasi untuk memilih kromosom-kromosom dari populasi yang akan menjadi orang tua bagi generasi berikutnya. Menurut Sri Kusumadewi[15] ,langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu tempat seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif individu tersebut terhadap semua individu dalam tempat seleksi tersebut. Metode *Rank Selection* merupakan salah satu metode seleksi yang dimiliki algoritma genetika. Metode *Rank Selection* bekerja dengan terlebih dahulu me-*ranking* atau mengurutkan kromosom di dalam populasi berdasarkan *fitness*-nya. Kemudian memberi nilai *fitness* baru berdasarkan urutannya. Kromosom dengan *fitness* terburuk akan memiliki *fitness* baru yang bernilai 1, terburuk kedua bernilai 2, dan seterusnya. Sehingga kromosom yang memiliki *fitness* terbaik akan memiliki nilai *fitness* yang sama dengan adalah jumlah kromosom dalam sebuah populasi. Setelah proses pengurutan dan pemberian nilai *fitness* baru, setiap kromosom akan memiliki kesempatan yang lebih kecil untuk terpilih, sehingga metode ini menyebabkan konvergensi menjadi lambat.

1.10 Crossover

Pindah silang *(crossover)* Pindah silang *(crossover)* merupakan komponen paling penting dalam algoritma genetika dan *memetika* karena proses inilah yang dapat mendekati kromosom-kromosom sebagai kandidat solusi yang diharapkan. Kromosom yang mengarah pada nilai penalti yang lebih bagus diperoleh dari proses memindah silangkan dua buah kromosom yang diperoleh dari seleksi orang tua. Pindah silang dapat menghasilkan satu atau beberapa kromosom yang biasa disebut kromosom anak. Pindah silang dapat berakibat buruk jika jumlah populasinya sangat kecil karena suatu kromosom dengan gen-gen yang mengarah ke solusi (kromosom yang diharapkan) akan sangat cepat menyebar ke kromosom-kromosom lainnya. Untuk mengatasi masalah tersebut pindah silang hanya dapat dilakukan untuk probabilitas tertentu. Pindah silang dilakukan jika suatu bilangan random (0,1) yang dibangkitkan kurang dari yang ditentukan. Pada umumnya, ditentukan mendekati satu, misal 0,8. Menurut Suhartono E[8], nilai probabilitas *crossover* berkisar antara 85% hingga 90%. Menurut Suyanto [16], pindah silang yang paling sederhana adalah *one point crossover*. Dalam *one point crossover* dua titik dipilih secara random untuk membagi orang tua yang pertama, kemudian gen pertama hingga titik yang terpilih diturunkan ke kromosom anak sesuai posisinya, sedangkan orang tua yang lainnya diperiksa, jika terdapat nilai gen yang belum ada pada anak, maka ditambahkan ke kromosom anak dengan cara *left-to-right*.

1.11 Mutasi

Mutasi merupakan proses pertukaran atau perubahan informasi yang dibawa oleh masing-masing gen dalam kromosom anak hasil pindah silang. Mutasi dilakukan pada beberapa gen dalam kromosom anak dengan probabilitas mutasi yang relatif kecil. Peluang mutasi didefinisikan sebagai persentase dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Apabila peluang mutasi terlalu besar maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya. Biasanya sebesar dengan *n* adalah jumlah gen dalam kromosom. Proses mutasi dikendalikan oleh nilai probabilitas mutasi. Menurut Suhartono E[8], untuk nilai probabilitas mutasi disarankan kecil antara 5% hingga 10%. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada *inisialisasi* populasi.

2 ISI PENELITIAN

Pada tahap ini akan membahas mengenai masalah yang akan dianalisis. Beberapa hal yang terdapat dalam tahap ini yaitu mengenai analisis masalah, analisis sistem yang akan dibangun serta metode yang digunakan untuk menyelesaikan masalah.

2.1 Analisis Masalah

Masalah penjadwalan kuliah merupakan permasalahan yang kompleks karena adanya berbagai komponen yang terdiri dari mata kuliah, dosen, ruang kelas, dan waktu kegiatan perkuliahan dengan memperhatikan batasan dan syarat tertentu. Dalam pembuatan jadwal terdapat beberapa batasan-batasan yang harus diperhatikan agar jadwal yang dibuat dapat digunakan. Di mana batasan-batasan tersebut tidak boleh ada yang dilanggar atau disebut *hard constraint*. Ada pula batasan-batasan yang boleh dilanggar atau disebut *soft constraint*.

Hard constraint (batasan keras) dalam penjadwalan Prodi IF Unikom adalah sebagai berikut :

a. Bentrok ruangan dan waktu mengajar, ketika jam, hari atau ruangan terisi dalam waktu yang sama.
b. Bentrok dosen, ketika dosen mengajar di jam yang sama, hari yang sama atau dosen yang sama.
c. Bentrok dengan waktu shalat jumat.
d. Bentrok waktu ketersediaan dosen khusus.

Soft constraint (batasan lunak) merupakan sebuah batasan di mana jika dilanggar tidak akan merusak jadwal, hanya akan mengurangi kualitas dari jadwal tersebut. *Soft constraint* didapatkan berdasarkan batasan dari suatu instansi. *Soft constraint* berpengaruh pada jadwal, ketika jumlah *soft constraint* yang dilanggar makin sedikit maka kualitas jadwal tersebut makin baik. Sedangkan *Soft constraint* (batasan lunak) yang ada ialah bentrok waktu ketersediaan dosen biasa.

2.2 Analisis Data Masukan

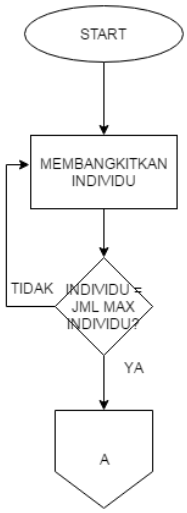
Data masukan yang digunakan adalah data dosen sebanyak 76 data, dosen mata kuliah sebanyak 325 data, mata kuliah sebanyak 25 data, ruang sebanyak 18 data, dan *slot* waktu sebanyak 120 *slot*.

2.3 Analisis Metode

Analisis metode adalah proses menjabarkan atau menjelaskan tahapan-tahapan dari algoritma memetika. Adapun tahapannya dari pembentukan kromosom, cek nilai *fitness*, proses seleksi, proses *crossover*, proses mutasi, dan proses pencarian lokal (*local search*). Semua tahapan tersebut dilakukan sebanyak generasi yang diinginkan oleh pengguna. Setelah semua tahapan dilalui terdapat beberapa kondisi yaitu solusi ditemukan, dan solusi tidak ditemukan setelah mencapai generasi terakhir.

2.3.1 Pembangkitan Kromosom

Pembangkitan kromosom adalah proses pembentukan kromosom yang nantinya akan menjadi suatu solusi (jadwal). Kromosom ini terdiri dari gen-gen yang isinya Dosen Mata kuliah, Jam, Hari, Ruang. Individu dibangkitkan sesuai dengan data masukan oleh pengguna. Makin banyak individu yang di masukan oleh pengguna maka semakin banyak juga calon-calon solusi yang dihasilkan dan makin banyak juga perbedaan antar individu yang memungkinkan ditemukannya solusi dengan cepat. Jumlah populasi yang harus dimasukkan harus bernilai genap. Di dalam satu individu terdapat kromosom yang jumlahnya sebanyak dosen mata kuliah (bisa dilihat di gambar 3.4 dan lengkapnya di lampiran data dosen mata kuliah). Pemberian nilai-nilai ke dalam gen tersebut dilakukan secara acak. Kecuali untuk gen mata kuliah, nilainya diambil secara berurutan. Alur dari proses pembangkitan adalah sebagai berikut (gambar 1):



Gambar 1. *Flowchart* Proses Pembangkitan.

2.3.2 Cek *Constraint* dan Nilai *Fitness*

Pada tahap ini individu yang sudah dibangkitkan akan di cek *constraint* yang dilanggar dan nilai *fitness*-nya. Di mana nilai *fitness* dipengaruhi oleh banyaknya pelanggaran yang ada dalam masing-masing individu. Untuk setiap pelanggaran maka akan dikenakan penalti +1.

Untuk *Hard constraint* yang dianalisis adalah sebagai berikut :

1. Bentrok ruang, hari dan jam yang sama (penalti). Maksudnya ketika ada jadwal yang memiliki jam yang sama, hari yang sama, dan ruang yang sama. *Constraint* ini dapat dibagi lagi :
   a. Ketika sks >=2 dan jam kedua = jam 1 mata kuliah lain ditambah hari dan ruangan yang sama.

b. Ketika sks >= 3 dan jam ketiga = jam 1 mata kuliah lain ditambah hari dan ruangan yang sama.
c. Ketika sks >=4 dan jam keempat = jam 1 mata kuliah lain ditambah hari dan ruangan yang sama.

2. Bentrok dosen, ketika dosen yang sama mengajar di jam dan hari yang sama(penalti2). *Constraint* ini dapat dibagi lagi :
   a. Ketika sks >=2 dan jam kedua = jam 1 mata kuliah lain ditambah hari yang sama.
   b. Ketika sks >=3 dan jam ketiga = jam 1 mata kuliah lain ditambah hari yang sama.
   c. Ketika sks >=4 dan jam keempat = jam 1 mata kuliah lain ditambah hari yang sama.

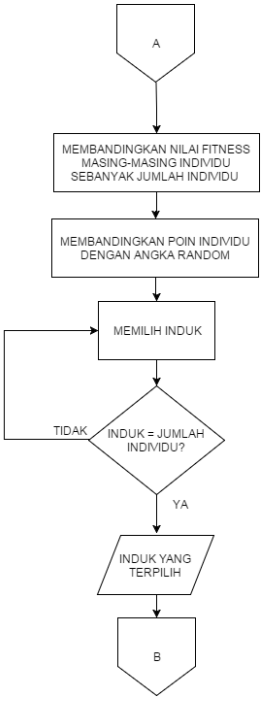
3. Bentrok shalat jumat(penalti3), ketika mata kuliah di jadwalkan di waktu shalat jumat. Waktu yang digunakan untuk shalat jumat bisa dilihat di (tabel 3.3). Slot waktu yang digunakan *slot* 7,8,9. *Constraint* ini dibagi lagi :
   a. Ketika sks >=2 dan jam kedua = jam 1 waktu shalat jumat.
   b. Ketika sks >=3 dan jam ketiga = jam 1 waktu shalat jumat.
   c. Ketika sks >=4 dan jam keempat = jam 1 waktu shalat jumat.

4. Bentrok waktu dengan keinginan dosen khusus(penalti4), ketika jam dan hari keinginan dosen sama dengan yang dijadwalkan.

5. Bentrok waktu dengan keinginan dosen khusus (penalti5), ketika jam dan hari keinginan dosen sama dengan yang dijadwalkan

2.3.3 Seleksi

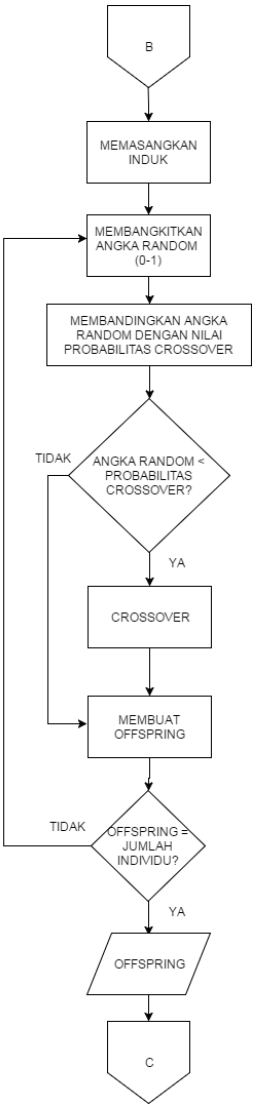
Proses seleksi adalah proses memilih individu yang akan dijadikan induk berdasarkan *ranking* individu tersebut. Semakin besar *ranking* individu tersebut maka semakin besar kemungkinan untuk terpilih menjadi induk. *Ranking* didapat dengan membandingkan nilai *fitness* dari masing-masing individu. Nilai dari masing-masing individu tersebut diambil dari menambahkan poin kepada setiap individu setelah dibandingkan dengan individu yang lain, semakin besar nilai mendekati jumlah populasi maka semakin besar pula nilai individu tersebut. Individu dengan poin sama dengan 1 merupakan individu dengan *rank* terkecil. Pemilihan individunya dilakukan pertama dengan membangkitkan angka random rentangnya antara (1 sampai dengan banyak individu) lalu dibandingkan dengan poin masing-masing individu. Sehingga individu dengan poin yang besar memiliki kemungkinan terpilih lebih besar. Proses ini dilakukan sampai induk berjumlah sama dengan jumlah individu. Setelah proses me-*ranking* tersebut maka didapatkan induk.



Gambar 2. *Flowchart* Proses Seleksi.

2.3.4 *Crossover*

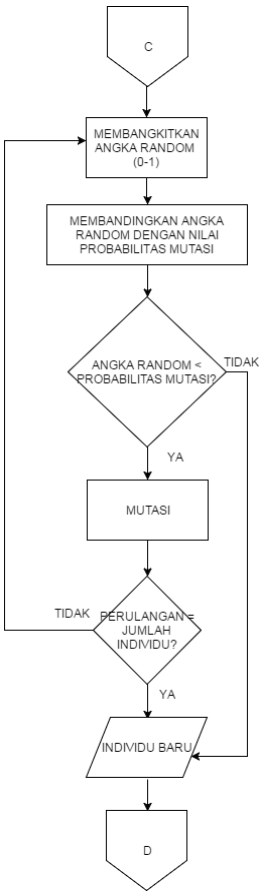
Proses *crossover* adalah proses menukarkan beberapa gen induk dengan gen induk lain. Proses ini dilakukan ketika nilai random (0 sampai 1) kurang dari probabilitas *crossover* yang telah ditentukan (0 sampai 1). Ketika angka random lebih dari probabilitas maka tidak dilakukan proses *crossover* dan *offspring* yang di ciptakan gennya sama dengan gen induk. Gen yang ditukar yaitu gen jam, hari, dan ruang. Proses ini dilakukan harapannya menghasilkan individu baru yang lebih baik dari individu sebelumnya.



Gambar 3. *Flowchart* Proses *Crossover*.

2.3.5 Mutasi

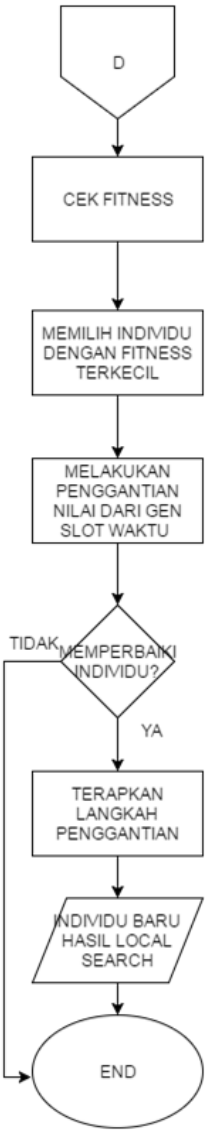
Proses mutasi adalah proses mengganti isi dari gen di satu kromosom dalam satu individu. Gen yang diubah nilainya adalah gen jam, hari, dan ruangan. Pergantian nilainya dilakukan secara random. Proses ini dilakukan ketika nilai random (0-1) kurang dari probabilitas mutasi yang telah ditentukan(0-1).



Gambar 4. Flowchart Proses Mutasi.

2.3.6 Pencarian Lokal

Proses pencarian lokal yang digunakan adalah dengan mengganti informasi yang kurang tepat di dalam satu individu yang memiliki penalti terbesar. Prosesnya dengan mengganti *slot* waktu. *Slot* waktu yang diganti hanya *slot* waktu kromosom memiliki penalti, dimaksudkan agar kualitas dari individu tetap baik.



Gambar 5. *Flowchart* Proses Pencarian Lokal.

2.4 Hasil Pengujian

Pengujian parameter algoritma dilakukan untuk mengamati bagaimana pengaruh probabilitas crossover, dan probabilitas mutasi terhadap jumlah generasi dan waktu yang diperlukan untuk menemukan solusi. Parameter yang akan digunakan sesuai dengan rencana pengujian di tabel 4.4. Parameter - parameter tersebut akan diuji hingga mencapai nilai fitness mencapai 1 (tidak ada constraint yang dilanggar). Pengujian dilakukan sebanyak 10x pada masing-masing nilai paramater. Dengan tujuan untuk mendapatkan berapa ukuran parameter yang cocok untuk menemukan solusi jadwal yang optimal.

2.4.1 Pengujian Parameter *Crossover*

Pengujian parameter *Crossover* dilakukan sebanyak 10x dan didapatkan hasilnya pada Tabel 1.

Tabel 1. Tabel Pengujian Parameter *Crossover.*

Parameter	**Nilai P *Crossover***	
	0.90	**0.95**
Waktu yang dibutuhkan (menit)	188.7 menit	**158.2 menit**
Generasi	16822	**12029**
Rata-Rata Nilai Fitness	0.3075	**0.3164**

Hasil dari pengujian didapatkan bahwa probabilitas *crossover* dengan nilai 0.95 lebih baik dibandingkan dengan 0.90 dalam sisi waktu, generasi hingga mencapai solusi optimal, dan rata-rata nilai fitness yang dihasilkan.

2.4.2 Pengujian Parameter Mutasi

Berdasarkan dari pengujian probabilitas crossover sebelumnya, maka nilai probabilitas crossover yang digunakan adalah 0.95. Pengujian parameter Mutasi dilakukan sebanyak 10x dan didapatkan hasilnya pada Tabel 2.

Tabel 2. Tabel Pengujian Parameter Mutasi.

Parameter	**Nilai P Mutasi**	
	0.05	**0.10**
Waktu yang dibutuhkan (menit)	140.7 menit	**118 menit**
Generasi	11325	**9971**
Rata-Rata Nilai Fitness	**0.3164**	0.3118

Hasil dari pengujian didapatkan bahwa probabilitas mutasi dengan nilai 0.10 lebih baik dibandingkan dengan 0.05 dalam sisi waktu, dan generasi hingga mencapai solusi optimal, tetapi tidak dengan rata-rata nilai fitness yang dihasilkan. Karena selisih rata-rata fitness yang tidak terlalu jauh maka diambil 0.10 untuk nilai probabilitas mutasi.

2.4.3 Pengujian *Soft Constraint*

Soft constraint yang digunakan adalah ketersediaan dosen biasa, Di mana diterapkan kepada dosen secara bertahap, mulai dari 1 dosen hingga 5 dosen. Semua dosen tersebut memiliki waktu tidak bersedia sebanyak 100 slot di dalam *slot* waktu. Pengujian menggunakan parameter dengan nilai probabilitas *crossover* 0.95, probabilitas mutasi 0.10, generasi sebanyak 1000000, dan populasi sebanyak 18. Setelah dilakukan percobaan. Maka didapatkan hasil di tabel 3.

Tabel 3. Tabel Pengujian *Soft Constraint.*

Percobaan	**Jumlah Dosen**	**Yang Dilanggar**
1	1	2
2	2	5
3	3	8
4	4	15
5	5	17

Bisa dilihat nilai dari *soft constraint* yang dilanggar setiap percobaan mengalami peningkatan. Maka didapatkan kesimpulan ketika jumlah dosen yang memiliki *soft constraint* semakin banyak maka *soft constraint* yang dilanggar pun akan semakin banyak. Artinya jadwal yang terbentuk masih terdapat *soft constraint* sehingga jadwal belum dikatakan optimal sepenuhnya.

2.5 Kesimpulan Pengujian

Berdasarkan hasil pengujian yang ditampilkan pada tabel 1 dan 2 didapatkan parameter yang optimal untuk menyelesaikan permasalahan penjadwalan dengan data sebanyak 325 yaitu nilai probabilitas *crossover* sebesar 0.95 dan probabilitas mutasi sebesar 0.10. Dengan jumlah populasi 18 dan generasi sebanyak 1000000000. Dan berdasarkan tabel 3 didapatkan bahwa semakin banyak jumlah dosen yang memiliki *soft constraint* maka semakin banyak pula jumlah *soft constraint* yang dilanggar dan ini membuat jadwal kurang optimal seluhurnya.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil yang didapat pada penelitian ini, diperoleh informasi penggunaan algoritma *memetika* dapat menyelesaikan permasalahan penjadwalan dengan data yang banyak (tidak ada *hard constraint* yang dilanggar atau nilai *fitness* sama dengan 1). Tetapi jadwal yang dihasilkan masih terdapat *soft constraint* sehingga jadwal belum bisa dikatakan optimal sepenuhnya. Proses pencarian lokal sangat membantu sekali dalam peningkatan kualitas dari individu atau peningkatan nilai *fitness* tiap generasinya. Proses pencarian lokal juga dapat mengatasi nilai lokal optimal yang sering terjadi di algoritma genetika ketika data masukannya terbilang sangat banyak. Hasil uji dari beberapa parameter didapatkan bahwa parameter dengan nilai probabilitas *crossover* sebesar 0.95 dan mutasi sebesar 0.10 akan mengoptimalkan proses pencarian solusi optimal. Didapatkan juga, ketika jumlah dosen yang memiliki *soft constraint* semakin banyak, maka *soft constraint* yang dilanggar pun akan semakin banyak. Hal tersebut mengakibatkan kualitas dari jadwal menjadi berkurang.

3.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, masih perlu dilakukan beberapa kajian lebih lanjut. Adapun saran untuk penelitian lebih lanjut yaitu:

1. Menggunakan jumlah populasi yang lebih banyak lagi, sehingga kemungkinan mencapai solusi optimal lebih cepat.
2. Mengoptimalkan proses pencarian lokal agar lebih cepat menyelesaikan permasalahan penjadwalan dengan data banyak.
3. Melakukan penanggulangan *soft constraint*, agar jadwal yang dihasilkan benar-benar optimal sepenuhnya.

DAFTAR PUSTAKA

[1] Albar Ali M. 2013. Algoritma Genetik Tabu Search dan Memetika Pada Permasalahan Penjadwalan.Seminar Nasional Teknologi Informasi dan Multimedia 2013.

[2] Sadaf N. Jat, Shengxiang Yang. 2011. A Memetic Algorithm for the University Course Timetabling Problem. Systems Man and Cybernetics Part C: Applications and Reviews IEEE Transactions on, vol. 41, pp. 93-106, ISSN 1094-6977.

[3] Garg Poonam. 2009. Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm,International Journal of Network Security & Its Applications (IJNSA), Vol.1, No 1.

[4] Ray Pillarrichie, Suyanto. 2012. Algoritma Genetika Dengan Local Search Untuk Penjadwalan Kuliah, Skripsi S1, Fakultas Teknik Informatika, Telkom University.

[5] Yamada, T., Reeves, C. 1997. Permutation flowshop Schedulling by genetic local search. IEEE Conference Publication. 232-238. IEEE.

[6] Rossi-Doria Olivia, Paetcher Ben. 2004. A memetic algorithm for University Course Timetabling. Proc. of Combinatorial Optimisation.

[7] David E. Goldberg. 1989. Genetic Algorithms in Search, Optimization & Machine Learning,Addison- Wesley.

[8] Suhartono E. 2015. Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika. Jurnal Ilmiah INFOKAM. No II/ Th. XI/ Sept/ 15.