

IMPLEMENTASI ALGORITMA GENETIKA PADA PENJADWALAN PERKULIAHAN

Desti Arghina Listyaningrum¹, Uning Lestari², Naniek Widayastuti³

^{1,2,3} Teknik Informatika, FTI, IST AKPRIND

¹destiarghina@gmail.com, ²uning@akprind.ac.id, ³naniek_wid@yahoo.com

ABSTRACT

In academic information system, scheduling is one of the allocation problem of academic activity in certain time. The solution of academic scheduling in large scale still face up so many obstacles to done manually. The college has to give a schedule in certain time when every academic activity is not crash. Scheduling is needed to anticipate crash of students hours to study and lectures time to teach. Scheduling have to fill the boundary and condition so that it convenient when it used. Based on the condition, a system is needed to arrange schedule will not crash so it can improve everyone's jobs. The possibility to find the best result and the implemented method approach to solve the problem use genetic algorithm method. Making this Scheduling application is started with build population of combination from class data, room data, lecture data and time slot and followed by initiation and calculation with genetic algorithm. During the process of algorithm calculation, chromosome check is also must done if there is a crash. From the result, indicate fine schedule means there is no crash between each other and all class can be scheduled. The schedule is optimal if it fill the condition which is process will stop if the solution is got or there is no solution. So, genetic algorithm can be implemented in application to make fine and optimal schedule.

Keywords: schedulling, academic, genetic algorithm

INTISARI

Dalam sistem akademik perguruan tinggi, penjadwalan merupakan salah satu permasalahan pengalokasian aktivitas perkuliahan ke dalam slot waktu yang telah ditentukan. Penyelesaian masalah penjadwalan perkuliahan dalam jumlah yang sangat besar hingga saat ini masih menjadi permasalahan yang rumit untuk diselesaikan secara manual. Perguruan Tinggi harus memberikan jadwal yang nantinya masuk kedalam waktu tertentu dimana setiap perkuliahan tidak benturan. Penjadwalan pada umumnya diperlukan untuk mengantisipasi adanya benturan jam kuliah dan juga waktu dosen dalam mengajar. Jadwal yang dihasilkan juga harus memenuhi batasan dan syarat yang bertujuan agar jadwal yang dihasilkan sesuai saat digunakan. Berdasarkan kondisi diatas, maka dibutuhkan suatu sistem untuk membuat jadwal supaya tidak benturan sehingga bisa meningkatkan efisiensi kerja dari berbagai pihak. Kemungkinan yang terjadi untuk menemukan hasil terbaik, dan metode yang diterapkan untuk menyelesaikan masalah penjadwalan perkuliahan. Salah satu metode yang dapat digunakan adalah pendekatan menggunakan Algoritma Genetika. Proses aplikasi penjadwalan ini dimulai dengan membangun populasi yang merupakan kombinasi dari data kelas, data ruang, data dosen dan time slot, lalu dilanjutkan dengan inialisasi awal dan perhitungan menggunakan algoritma genetika. Pada proses perhitungan algoritma juga dilakukan pengecekan chromosome jika terjadi benturan. Dari hasil yang ada, diperoleh hasil jadwal yang baik yang berarti tidak ada benturan antar jadwal dan semua kelas dapat dijadwalkan. Jadwal yang dihasilkan juga dikatakan optimal jika memenuhi syarat yang ada yaitu proses akan berhenti jika solusi sudah didapat atau tidak pernah ada solusi. Jadi Algoritma Genetika, yang diterapkan pada aplikasi dapat menghasilkan jadwal yang baik dan optimal.

Kata Kunci : penjadwalan, perkuliahan, algoritma genetika

PENDAHULUAN

Jadwal terhadap suatu kegiatan merupakan hal yang penting dilakukan agar kegiatan tersebut berlangsung secara lancar. Penyusunan jadwal kegiatan berkaitan dengan berbagai syarat yang harus dipenuhi sehingga memerlukan banyak pertimbangan untuk mendukung kegiatan tersebut.

Penjadwalan perkuliahan merupakan masalah penempatan jadwal suatu aktivitas kuliah pada waktu, kelas dan ruang dan dosen yang telah ditentukan. Ada dua nilai benturan pertama yaitu benturan ruang adalah dalam satu chromosome ada dua ruang dalam jam yang sama. Sedangkan benturan kedua yaitu dalam satu chromosome ada dua kelas yang diampu dosen yang sama pada jam yang sama. Penyelesaian masalah penjadwalan perkuliahan dalam jumlah yang sangat besar hingga saat ini masih menjadi permasalahan yang rumit untuk diselesaikan secara manual.

Penjadwalan pada umumnya diperlukan untuk mengantisipasi adanya benturan jam kuliah dan juga waktu dosen dalam mengajar. Jadwal yang dihasilkan juga harus memenuhi batasan dan syarat yang bertujuan agar jadwal yang dihasilkan sesuai saat digunakan.

Maka, dari permasalahan tersebut, dibutuhkan suatu sistem untuk membuat jadwal supaya tidak benturan sehingga bisa meningkatkan efisiensi kerja dari berbagai pihak. Kemungkinan yang terjadi untuk menemukan hasil terbaik, dan metode yang diterapkan untuk menyelesaikan masalah penjadwalan perkuliahan.

Berdasarkan penjabaran di atas, maka dapat dirumuskan beberapa permasalahan antara lain: Bagaimana membuat sebuah perangkat lunak yang menyelesaikan masalah penjadwalan perkuliahan dengan Algoritma Genetika tanpa adanya benturan.

Penelitian ini bertujuan untuk membangun aplikasi pengujian penjadwalan perkuliahan menggunakan Algoritma Genetika dan aplikasi yang dibangun menggunakan Algoritma Genetika.

TINJAUAN PUSTAKA

Optimasi penjadwalan perkuliahan. Metode yang diterapkan adalah Algoritma Genetika ini digunakan untuk menyelesaikan masalah penjadwalan perkuliahan sehingga diperoleh hasil yang optimal (Ulfa, 2011). Kelebihan dari aplikasi yaitu menghasilkan penjadwalan yang optimal dengan tingkat kesalahan yang kecil. Dan kekurangan dari aplikasi ini yaitu masih terjadi benturan saat penjadwalan sehingga masalah penjadwalan belum selesai karena masih terjadi benturan.

(Shadid, 2008) Aplikasi tentang menyempurnakan aplikasi penyusunan jadwal kuliah dari penelitian sebelumnya. Metode yang diterapkan adalah Algoritma Genetika. Algoritma Genetika dapat memberikan hasil yang optimal dalam masalah penjadwalan. Kelebihan dari aplikasi yaitu kinerja dari Algoritma Genetika mencapai nilai optimal. Dan kekurangan dari aplikasi ini belum mengakomodasi atau mempercepat penerapan Algoritma Genetika.

(Fachrudin, 2010) Aplikasi untuk menyelesaikan masalah operasi tugas pada kinerja mesin. Algoritma yang digunakan Algoritma Genetika untuk mengoptimalkan cara kerja mesin dalam beroperasi. Kelebihan dari Algoritma Genetika dapat dijalankan pada berbagai order dan efisiensi operasi yang sudah ditentukan. Kekurangan dari aplikasi ini yaitu memerlukan waktu lama dalam mengoperasikan algoritma genetika.

(Fitri, 2004) Sistem penjadwalan perkuliahan dengan algoritma genetika. Kelebihan sistem yaitu hasil yang dicapai dengan kromosom terbaik adalah yang bernilai satu. Kekurangan kapasitas penyimpanan sangat kecil.

METODOLOGI PENELITIAN

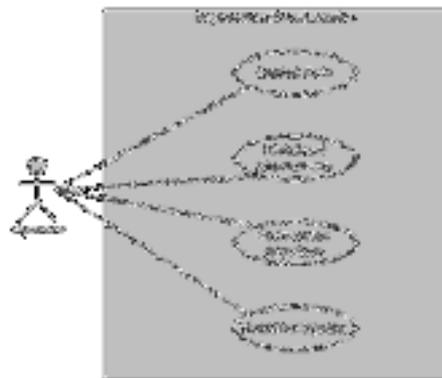
Langkah penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Menganalisis dan mengidentifikasi alat dan bahan yang diperlukan dalam penelitian.
2. Merancang *database* dan *design* sistem.
3. Melakukan pembuatan *source code* sistem sesuai dengan *database* dan *design* yang dirancang.

4. Melakukan pengujian terhadap sistem.
5. Melakukan pencarian *error* yang terjadi pada sistem.
6. Evaluasi chromosome, menghitung nilai *fitness* rumus yang digunakan $Fitness [i] = \text{Jumlah Benturan Ruang} + \text{Jumlah Benturan Dosen}$.
7. Seleksi chromosome, menghitung nilai *inverse* dari hasil evaluasi chromosome rumus yang digunakan $Q[i] = 1/Fitness [i]$. Hitung nilai probabilitas menggunakan rumus $P[i] = Q[i]/\text{Total } inverse$, proses seleksi menggunakan *roulette-wheel* tetapi harus menghitung nilai kumulatif probabilitas terlebih dahulu dengan menjumlahkan semua hasil probabilitas. Lalu membangkitkan bilangan acak R untuk menentukan induk chromosome.
8. *Crossover* dihitung menggunakan *probability crossover* sebesar 25%. Dengan membangkitkan bilangan acak R sebanyak populasi. Chromosome yang akan dipilih adalah chromosome dengan nilai acak yang lebih kecil dari 25%. Chromosome itu yang nantinya akan di *crossover* dengan menentukan *cut-point crossover* yaitu sub-chromosome mana yang akan di *crossover*. Setelah proses *crossover* selesai urutan semua chromosome dari pertama sampai terakhir termasuk chromosome yang sudah di *crossover*.
9. Mutasi, menghitung panjang total sub-chromosome. Jumlah sub-chromosome adalah 50 dan jumlah populasi atau jumlah chromosome adalah 10, dengan rumus $\text{Total Sub-Chromosome} = \text{Jumlah Sub-Chromosome} * \text{jumlah populasi}$. Untuk memilih posisi sub-chromosome yang mengalami mutasi dengan cara membangkitkan nilai acak antara 1 – total sub-chromosome. Dengan membangkitkan *mutation rate* 10% (0.1) dari total sub-chromosome, rumus yang digunakan $\text{Jumlah Mutasi} = 0.1 * \text{Total Sub-Chromosome}$.

Use Case Diagram

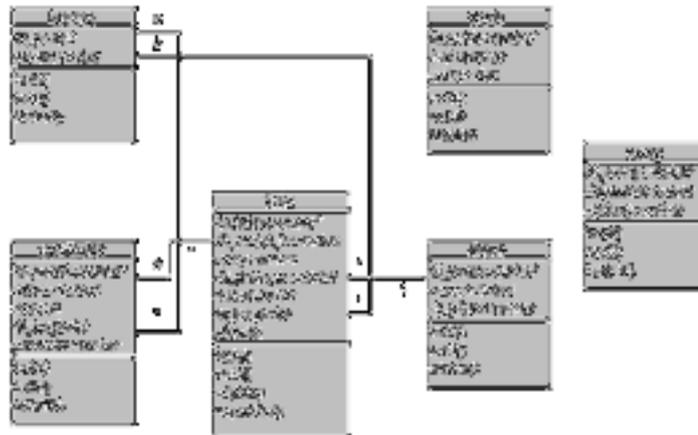
Pada Gambar 1 Use Case Diagram, berikut terdapat aktor yaitu operator. Pada implementasi penjadwalan ini terdapat alur pada sistem yang akan digunakan. Operator melakukan beberapa aktivitas untuk menjalankan sistem. Aktivitas yang dilakukan operator adalah melihat data, inialisasi kromosom, perhitungan algoritma, dan cetak laporan.



Gambar 1 Use Case Diagram

Class Diagram

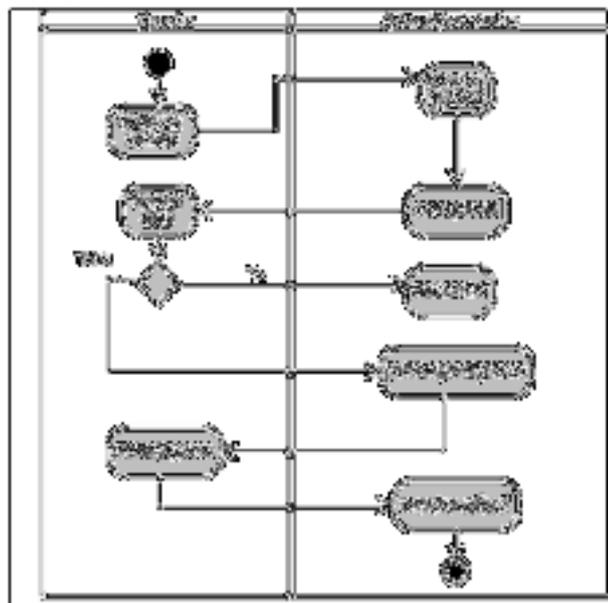
Pada *Class Diagram* sesuai Gambar 2 Class Diagram menunjukkan hubungan dari masing-masing *class* ke *class* lain. Terdapat 6 *class* yaitu jurusan, dosen, matakuliah, kelas, waktu dan ruang.



Gambar 2 Class Diagram

Activity Diagram

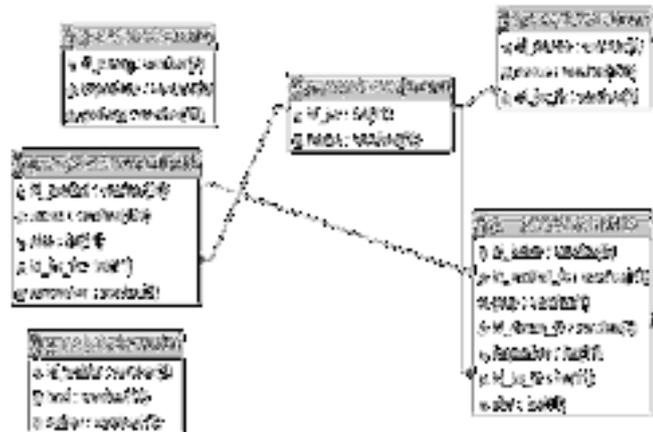
Pada Gambar 4 Activity Diagram. *Activity Diagram Operator*, disini akan terlihat lebih spesifik apa saja yang dilakukan operator untuk mengerjakan sistem penjadwalan. Operator akan membuka aplikasi penjadwalan akan muncul halaman aplikasi, operator akan memilih menu yang dikehendaki. Terdapat tiga menu, menu pertama yaitu master data pada master data terdapat data kelas, data ruang, data time slot dan data dosen, menu kedua yaitu operator menghitung inialisasi chromosome yang digunakan dan menghitung dengan algoritma disini operator bisa mereset jika ingin mengulanginya, menu ketiga yaitu operator mencetak hasil perhitungan chromosome dan perhitungan secara menyeluruh. Jika proses pada aplikasi sudah berakhir operator akan menutup aplikasi dan proses penjadwalan selesai.



Gambar 4 Activity Diagram

Relasi Database

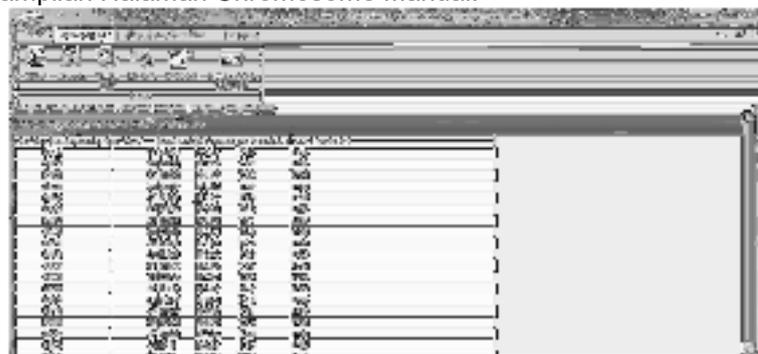
Dari semua table pada aplikasi penjadwalan, maka dapat direlasikan untuk table-table tersebut pada Gambar 5 Relasi Database di bawah ini.



Gambar 5 Relasi Database

PEMBAHASAN
Pengujian Sistem Manual
Chromosome Manual

Pada halaman ini memuat data kromosom manual yang akan diujikan menggunakan algoritma genetika. Data chromosome awal manual dapat dilihat pada Gambar 6 Tampilan Halaman Chromosome Manual.

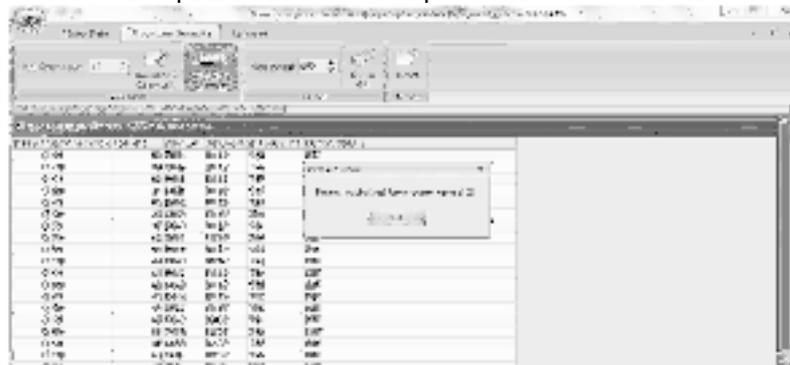


Gambar 6 Tampilan Halaman Chromosome Manual

Dari gambar diatas terdapat 10 chromosome yang akan diinisialisasi awal pada tahapan perhitungan algoritma genetika.

Inisialisasi Awal Manual

Pilih halaman algoritma genetika maka akan muncul halaman ini. Pilih inisialisasi manual lalu akan muncul pada Gambar 7 Tampilan Halaman Inisialisasi Awal Manual.



Gambar 7 Tampilan Halaman Inisialisasi Awal Manual

Setelah proses inialisasi awal manual selesai akan keluar data kromosom awal manual yang sudah di inialisasi. Hasil ini dapat dilihat pada Gambar 8 Tampilan Halaman Hasil Inialisasi Manual.

Chromosom	1	2	3	4	5
1	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000	0.0000	0.0000
28	0.0000	0.0000	0.0000	0.0000	0.0000
29	0.0000	0.0000	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000	0.0000	0.0000

Gambar 8 Tampilan Halaman Hasil Inialisasi Manual

Dari proses diatas didapat hasil inialisasi awal dari sistem manual yang akan dihitung menggunakan algoritma genetika.

Perhitungan Algoritma Genetika Sistem Manual

Pada halaman ini akan diujikan kromosom awal manual yang tadi sudah diinialisasikan. Pilih iterasi untuk menghitung nilai algoritma genetika yang akan dilakukan berapa kali. Disini menggunakan iterasi 31 yaitu berulang menghitung sebanyak 31 kali dapat dilihat pada Gambar 9 Tampilan Halaman Pilih Iterasi.

Iterasi	1	2	3	4	5
1	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000	0.0000	0.0000
28	0.0000	0.0000	0.0000	0.0000	0.0000
29	0.0000	0.0000	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0000	0.0000	0.0000

Gambar 9 Tampilan Halaman Pilih Iterasi

Lalu pada perhitungan masuk GA. Diperoleh hasil yang dapat dilihat pada Gambar 10 Tampilan Halaman Hasil Perhitungan.

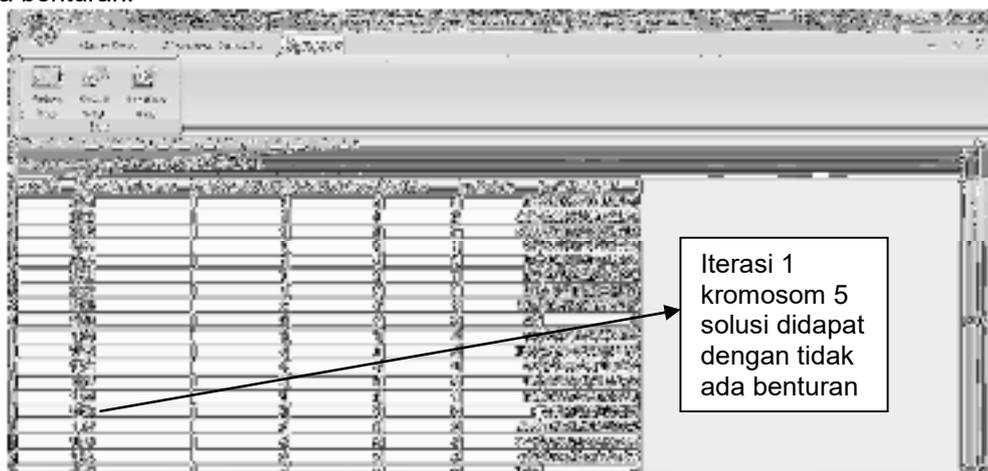


Gambar 10 Tampilan Halaman Hasil Perhitungan

Dari perhitungan algoritma genetika diatas didapatkan solusi, solusi yaitu hasil perhitungan yang sudah dilakukan menghasilkan jadwal yang diinginkan yaitu tidak ada benturan dosen, tidak ada benturan ruang, syarat *fitness* = 0 terpenuhi dan nilai peluang 1. Jika dalam nilai peluang didapat nilai 1 maka mendapat solusi yang berarti tidak ada benturan, sedangkan jika dalam nilai peluang tidak didapat nilai 1 maka tidak mendapat solusi yang berarti terjadi benturan. Pada percobaan I didapat solusi, ditemukan pada iterasi 1 pada kromosom ke-5.

Laporan Perhitungan Sistem Manual

Pada halaman laporan ini berisi laporan perhitungan yang sudah dilakukan. Laporan hasil perhitungan algoritma genetika dapat dilihat pada Gambar 11 Tampilan Halaman Laporan Rekap Log. Halaman ini memuat laporan perhitungan setiap kromosom yang sudah dilakukan. Di dalam halaman ini bisa diketahui perhitungan menemukan solusi atau tidak. Didapat solusi pada iterasi 1 kromosom 5 dengan benturan ruang 0, benturan dosen 0 dan nilai fitness 0 yang sudah memenuhi syarat sehingga tidak ada benturan.



Gambar 11 Tampilan Halaman Laporan Rekap Log

Pada hasil laporan perhitungan diatas didapat hasil yaitu pada perhitungan manual menggunakan 10 kromosom menggunakan 31 iterasi menghasilkan solusi pada iterasi 1 pada kromosom ke-5 yang berarti 10 percobaan menggunakan 31 kali perhitungan menghasilkan solusi pada perhitungan 1 dan percobaan ke-5. Arti dari jadwal diatas yaitu perhitungan algoritma genetika menghasilkan jadwal yang tidak ada benturan.

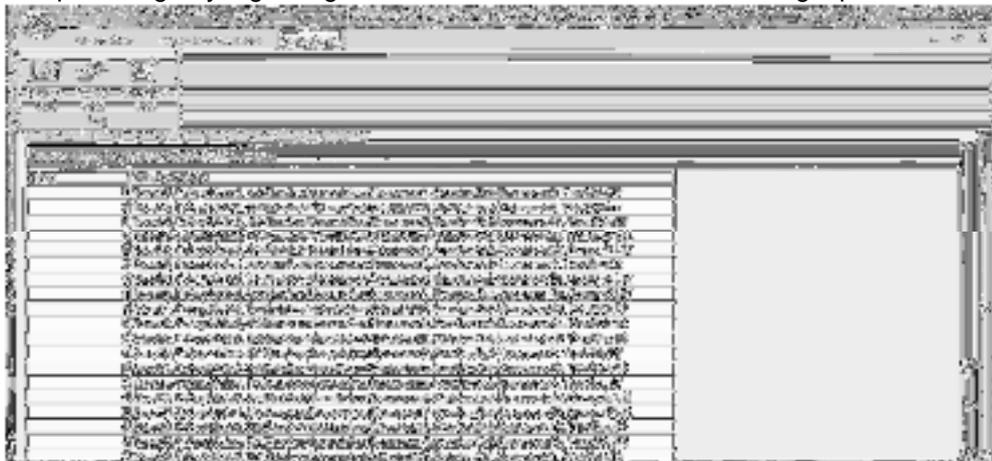
Pada Gambar 12 Tampilan Halaman Laporan Detail Log. Halaman ini memuat laporan perhitungan pada semua kromosom yang sudah dilakukan. Pada kolom keterangan terdapat keterangan perhitungan yang mengalami *crossover* dan mutasi.



Gambar 12 Tampilan Halaman Laporan Detail Log

Pada hasil laporan perhitungan diatas adalah hasil dari proses *crossover* dan mutasi, tetapi tidak semua proses perhitungan yang dilakukan melakukan proses *crossover* dan mutasi. Jika terjadi proses *crossover* dan mutasi dilakukan maka akan dicatat pada kolom keterangan yang menunjukkan proses tersebut.

Pada Gambar 13 Tampilan Halaman Laporan Random Log. Halaman ini memuat semua perhitungan yang mengalami *crossover* dan mutasi secara lengkap.



Gambar 13 Tampilan Halaman Laporan Random Log

Pada hasil laporan perhitungan diatas adalah hasil detail dari proses *crossover* dan mutasi. Hasil yang sebelumnya didapat yaitu pada laporan detail log akan dirinci pada laporan ini. Laporan ini akan menunjukkan pada iterasi dan kromosom berapa yang melalui tahapan dari proses *crossover* dan mutasi.

Pengujian Sistem Otomatis Inisialisasi Awal Otomatis

Pilih halaman algoritma genetika maka akan muncul halaman ini. Pilih jumlah chromosome yang akan dihitung. Pilih inisialisasi otomatis lalu akan muncul pada Gambar 14 Tampilan Halaman Inisialisasi Awal Otomatis.



Gambar 14 Tampilan Halaman Inisialisasi Awal Otomatis

Pada gambar diatas jumlah chromosome yang dipilih adalah 3 untuk menghitung berapa chromosome yang digunakan sesuai kebutuhan.

Setelah proses inisialisasi awal otomatis selesai akan keluar data kromosom awal otomatis yang sudah di inisialisasi. Hasil ini dapat dilihat pada Gambar 15 Tampilan Halaman Hasil Inisialisasi Otomatis.



Gambar 15 Tampilan Halaman Hasil Inisialisasi Otomatis

Pada gambar diatas chromosome berjumlah 4 chromosome ini merupakan hasil dari inisialisasi awal menggunakan 5 chromosome yang sudah dipilih yang akan dihitung menggunakan algoritma genetika.

Perhitungan Algoritma Genetika Sistem Otomatis

Pada halaman ini akan diujikan kromosom awal otomatis yang tadi sudah diinisialisasikan. Pilih iterasi untuk menghitung nilai algoritma genetika yang akan dilakukan berapa kali. Disini menggunakan iterasi 2 yaitu berulang menghitung sebanyak 2 kali dapat dilihat pada Gambar 16



Gambar 16 Tampilan Halaman Pilih Iterasi

Dari perhitungan algoritma genetika diatas tidak didapatkan solusi, solusi yaitu hasil perhitungan yang sudah dilakukan tidak menghasilkan jadwal yang diinginkan. Jika dalam nilai peluang didapat nilai 1 maka mendapat solusi yang berarti tidak ada benturan, sedangkan jika dalam nilai peluang tidak didapat nilai 1 maka tidak mendapat solusi yang berarti terjadi benturan. Pada percobaan II tidak didapat solusi.

Laporan Perhitungan Sistem Otomatis

Pada halaman laporan ini berisi laporan perhitungan yang sudah dilakukan. Laporan hasil perhitungan algoritma genetika dapat dilihat pada Gambar 18 Tampilan Halaman Laporan Rekap Log. Halaman ini memuat laporan perhitungan setiap kromosom yang sudah dilakukan. Di dalam halaman ini bisa diketahui perhitungan menemukan solusi atau tidak.

The image shows a screenshot of a software application window displaying a summary log report. The window has a title bar at the top. Below the title bar, there is a menu bar and a toolbar. The main area of the window contains a table with several columns and rows of data. The text is somewhat blurry but appears to be a structured list of results or logs.

Gambar 18 Tampilan Halaman Laporan Rekap Log

Pada hasil laporan perhitungan diatas tidak didapat hasil. Arti dari jadwal diatas yaitu perhitungan algoritma genetika menghasilkan jadwal berbenturan.

Pada Gambar 19 Tampilan Halaman Laporan Detail Log. Halaman ini memuat laporan perhitungan pada semua kromosom yang sudah dilakukan. Pada kolom keterangan terdapat keterangan perhitungan yang mengalami *crossover* dan mutasi.

The image shows a screenshot of a software application window displaying a detailed log report. The window has a title bar at the top. Below the title bar, there is a menu bar and a toolbar. The main area of the window contains a table with several columns and rows of data. In the center of the table, there is a small dialog box or pop-up window with a title bar and some text inside. The text is somewhat blurry but appears to be a structured list of results or logs.

Gambar 19 Tampilan Halaman Laporan Detail Log

Pada hasil laporan perhitungan diatas adalah hasil dari proses *crossover* dan mutasi, tetapi tidak semua proses perhitungan yang dilakukan tidak melakukan proses *crossover* dan mutasi. Jika terjadi proses *crossover* dan mutasi dilakukan maka akan dicatat pada kolom keterangan yang menunjukkan proses tersebut. Misal maksud dari iterasi 0 kromosom ke-0 idpart atau sub-chromosome 21 yaitu *crossover* dilakukan pada chromosome 2 dan chromosome 0 lalu di *cut-point crossover* menggunakan bilangan acak

11, setelah dilakukan *crossover* chromosome dimutasi pada sub-chromosome 0 yang akan dimutasi yaitu timeslot. Timeslot pada sub-chromosome 45 diganti menjadi T4. Penggantian jika terjadi kondisi mutasi yaitu jika sub-chromosome bernilai ganjil maka yang akan dimutasi adalah timeslot, sedangkan jika sub-chromosome bernilai genap maka yang akan dimutasi adalah ruang.

Pada Gambar 20 Tampilan Halaman Laporan Random Log. Halaman ini memuat semua perhitungan yang mengalami *crossover* dan mutasi secara lengkap.



Gambar 20 Tampilan Halaman Laporan Detail Log

Pada hasil laporan perhitungan diatas adalah hasil detail dari proses *crossover* dan mutasi. Hasil yang sebelumnya didapat yaitu pada laporan log akan dirinci pada laporan ini. Laporan ini akan menunjukkan pada iterasi dan kromosom berapa yang melalui tahapan dari proses *crossover* dan mutasi.

KESIMPULAN

Berdasarkan penjelasan, uraian pada bab-bab sebelumnya dan hasil penelitian, dapat diambil kesimpulan sebagai berikut :

1. Algoritma Genetika memperlihatkan hasil perhitungan yang sudah dilakukan.
2. Aplikasi ini hanya diujikan untuk kajian saja dengan satu sampel data.
3. Aplikasi hanya menggunakan satu sampel data yaitu data kelas yang berjumlah 50 data.
4. Pencarian solusi dilakukan dengan cara melakukan beberapa kali percobaan yang melibatkan beberapa nilai *probability crossover* (**pc**) dan nilai *mutation rate* (**pm**).
5. Perhitungan algoritma genetika dalam penjadwalkan perkuliahan dilakukan secara acak jadi hasil dari perhitungan tidak bisa ditentukan dan dipastikan menemukan solusi atau tidak menemukan solusi.
6. Hasil dari aplikasi hanya menampilkan hasil dari perhitungan nilai *probability crossover* (**pc**) dan nilai *mutation rate* (**pm**).
7. Durasi yang dibutuhkan dalam proses perhitungan algoritma genetika menyesuaikan pada data yang ada.

DAFTAR PUSTAKA

- Bangun, P. B. (2012). Penerapan Konsep Algoritma Genetika Untuk Penjadwalan Kegiatan Perkuliahan Semester Ganjil Kurikulum 2012 di Jurusan Matematika FMIPA UNSRI. *Jurnal Penelitian Sains* , 55-59.
- Fachrudin, A. (2010). *Penerapan Algoritma Genetika Untuk Masalah Penjadwalan Job Shop Pada Lingkungan Industri Pakaian* . Surabaya: Institut Teknologi Sepuluh Nopember.

- Fitri, R. (2004). *Penjadwalan Perkuliahan Dengan Pengujian Tabel Waktu (Time-Table) Menggunakan Algoritma Genetika*. 2004: Universitas Komputer Indonesia.
- Hermawanto, D. (2007). Tutorial Algoritma Genetika. *KIM - LIPI* , 1-7.
- Syadid, M. (2008). *Penjadwalan Perkuliahan Menggunakan Algoritma Genetika*. Bogor: Institut Pertanian Bogor.
- Ulfa, L. M. (2011). *Optimasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika*. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.
- Yunantara, M. D. (2012). Analisis dan Implementasi Penjadwalan Pengembangan Model Crossover Dalam Algoritma Genetika. *Jurnal Elektronik Ilmu Komputer - Universitas Udayana* , 16-17.