

ANALISIS DAN IMPLEMENTASI ALGORITMA GENETIKA PADA PENJADWALAN MATA PELAJARAN DI MADRASAH TSANAWIYAH MIFTAHUL 'ULUM DENPASAR

Ida Bagus Ary Indra Iswara¹, Dio Restu Saputra¹

¹Teknik Informatika, STMI STIKOM Indonesia
Denpasar, Indonesia

e-mail: indraiswara@stiki-indonesia.ac.id¹, diorestusaputra@gmail.com²

Abstrak

Penjadwalan pelajaran merupakan unsur penting dalam kegiatan belajar mengajar di sekolah. Penjadwalan pelajaran seringkali membutuhkan waktu yang cukup lama dikarenakan banyaknya faktor seperti penentuan alokasi jam mengajar guru, jumlah guru yang terbatas dan mengajar pada jumlah kelas yang tidak sedikit. Masalah yang sering muncul adalah bentrok jam mengajar di waktu yang sama. Salah satu solusi untuk permasalahan penjadwalan adalah melalui metode optimasi Algoritma Genetika. Algoritma Genetika merupakan metode yang terinspirasi dari teori genetika dan evolusi Mendell untuk menyelesaikan masalah yang membutuhkan optimasi. Batasan yang dijadikan constraint pada penelitian ini adalah bentrok jam mengajar guru dan bentrok kelas yang sama pada jam yang sama. Hasil dari implementasian Algoritma Genetika sebagai metode optimasi penjadwalan pelajaran belum mendapatkan hasil yang optimal. Solusi dengan nilai fitness terbaik dicapai dengan jumlah kromosom sebanyak 20 buah, jumlah generasi sebesar 200, nilai probabilitas crossover (Pc) 0,5 dan nilai probabilitas mutasi (Pm) bernilai dari 0,15 hingga 0,35. Waktu tercepat untuk nilai Pm 0,15 selama 91,22 detik, sedangkan pada nilai Pm 0,35, dibutuhkan waktu selama 91,65 detik.

Kata kunci: Penjadwalan, mata pelajaran, algoritma genetika, optimasi.

Abstract

Subject scheduling is an important element in teaching and learning activities in schools. Scheduling lessons often take a long time due to many factors such as determining teacher teaching hours allocation, the limited number of teachers and teaching in a large number of classes. The problem that often arises is the clash of teaching hours at the same time. One solution to scheduling problems is through the optimization method of Genetic Algorithms. Genetic Algorithms are methods inspired by Mendell's theory of genetics and evolution to solve problems that require optimization. The constraints that were used as constraints in this study were clashing hours of teaching teachers and clashing the same class at the same time. In this study, the results of implementing Genetic Algorithms as a method of optimizing lesson scheduling have not yet obtained optimal results. The solution with the best fitness value is achieved by the number of chromosomes as many as 20, the number of generations is 200, the probability of crossover (Pc) 0.5 and the probability value of mutation (Pm) is from 0.15 to 0.35. The fastest time for Pm value is 0.15 for 91.22 seconds, while for Pm value of 0.35, it takes 91.65 seconds.

Keywords : Scheduling, lesson, genetic algorithm, optimization.

PENDAHULUAN

Semakin berkembangnya teknologi informasi telah mempengaruhi perkembangan dunia pendidikan yang semakin meningkatkan pelayanannya, baik

dari segi kualitas guru, jumlah peserta didik, serta fasilitas yang mendukung proses pembelajaran. Salah satu hal yang membantu sekolah dalam mengatur proses pembelajaran adalah jadwal pelajaran.

Jadwal pelajaran (1)(2) adalah unsur penting yang mengatur proses pembelajaran, sedikit kesalahan saja dapat berakibat terjadinya tabrakan pertemuan. Seperti halnya Madrasah Tsanawiyah Miftahul 'Ulum Denpasar yang saat ini memiliki 15 kelas dan mengajarkan 16 mata pelajaran. Dengan jumlah pelajaran tersebut, maka diperlukan penyusunan jadwal pelajaran agar proses pembelajaran selama satu semester dapat berjalan dengan lancar.

Jadwal pelajaran pada Madrasah Tsanawiyah Miftahul 'Ulum Denpasar disusun oleh Ibu Wiji Utami, S.Pd selaku Waka Kurikulum di Madrasah Tsanawiyah Miftahul 'Ulum Denpasar.

Proses penyusunan jadwal tersebut dilakukan menggunakan aplikasi spreadsheet. Aplikasi tersebut menyimpan data guru, data siswa, data kelas, data mata pelajaran dan data jam pelajaran. Penyusunan jadwal pelajaran dilakukan dengan memasukkan kode guru dan kode mata pelajaran ke dalam tabel spreadsheet yang sudah diatur. Kekurangan pada sistem yang berjalan saat ini adalah waktu kerja yang cukup lama, sekitar 5-7 hari.

Hal ini disebabkan karena Waka Kurikulum memasukkan data ke dalam tabel dan memeriksanya satu per satu. Setiap ada jadwal yang sama dalam satu jam yang sama, maka beliau akan menukar jadwal dari hari lain secara acak. Begitu seterusnya hingga dicek kembali bahwa benar-benar sudah tidak ada jadwal yang bertabrakan. Kekurangan lain dari proses penjadwalan saat ini adalah akses informasi jadwal mengajar guru yang belum terintegrasi.

Berdasarkan uraian masalah tersebut, maka diperlukan sebuah sistem informasi. Menurut Susanto (3), sistem informasi adalah kumpulan dari sub sistem apapun baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengelola data menjadi informasi yang berarti dan berguna. Dari pengertian diatas, maka sistem informasi yang akan penulis bangun akan mengolah data berupa data guru, data kelas, data mata pelajaran dan data jam pelajaran. Data-data tersebut akan disusun secara otomatis

menggunakan metode optimasi algoritma genetika.

Fungsi algoritma genetika adalah untuk mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi (4) (5)(6). Algoritma genetika membantu untuk mendapatkan nilai terbaik dalam penyusunan jadwal yang telah dikelompokkan ke dalam tabel jadwal serta dapat mencegah terjadinya tabrakan jadwal pelajaran (7)(8). Dengan penggunaan algoritma tersebut, diharapkan sistem informasi ini dapat membantu Madrasah Tsanawiyah Miftahul 'Ulum Denpasar khususnya Ibu Wiji Utami, S.Pd selaku Waka Kurikulum dalam menyusun jadwal pelajaran menjadi lebih mudah dan cepat.

ALGORITMA GENETIKA

Algoritma genetika merupakan algoritma evolusioner untuk memecahkan masalah optimisasi berdasarkan pada ide-ide evolusi seleksi alam. Algoritma ini mensimulasikan proses evolusi dengan menghasilkan populasi solusi layak dan menerapkan beberapa operator genetik untuk menghasilkan populasi baru berdasarkan aturan seleksi (4).

Algoritma genetika dikembangkan lebih lanjut oleh John Holland bersama murid-murid dan rekan kerjanya dari Universitas Michigan sekitar tahun 1960-an dan 1970-an. John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi dapat diformulasikan dengan terminologi genetika. Sejak saat itu algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas di berbagai bidang. Pada awal tahun 1992 pemrograman mengenai genetika mulai diperkenalkan oleh J.H. Koza (9).

Algoritma genetika menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Dalam ilmu biologi, sekumpulan individu yang sama, hidup dan berkembang bersama dalam suatu area, disebut dengan populasi. Algoritma genetika bekerja dengan suatu populasi yang terdiri atas individu-individu, yang masing-masing individu mempresentasikan solusi yang mungkin bagi suatu permasalahan.

Algoritma genetika memiliki *performansi* yang baik untuk masalah-masalah selain optimisasi kompleks dari satu variabel atau multivariabel.

Salah satu aplikasi algoritma genetika adalah pada permasalahan optimisasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi, seperti optimisasi penjadwalan mata pelajaran. Pada algoritma genetika, kandidat solusi (yang selanjutnya disebut 'solusi') dari suatu masalah direpresentasikan sebagai kromosom.

Kromosom ini merupakan representasi dari solusi yang ada. Kumpulan kromosom disebut sebagai populasi. Masing-masing kromosom pada populasi akan dievaluasi menggunakan fungsi *fitness*, yaitu fungsi yang mengukur secara kuantitatif suatu kromosom untuk bertahan pada populasi. Pembentukan populasi baru dilakukan dengan menerapkan operator-operator genetika secara iteratif sampai dipenuhi kriteria berhenti, misalnya diperoleh populasi yang baik dari populasi sebelumnya.

Operator dasar dalam algoritma genetika adalah seleksi, *crossover* dan mutasi. Operator genetika *crossover* membentuk dua solusi (orang tua) dan menggabungkan dua solusi tersebut setelah satu atau lebih solusi (anak) dihasilkan. Orang tua yang dipilih merupakan orang tua dari semua solusi dalam populasi saat ini (9).

Algoritma genetika bekerja dengan menggunakan pendekatan random, sehingga nilai-nilai yang dihasilkan adalah nilai random. Pada kasus penjadwalan dengan model genetika, semakin banyak iterasi yang dilakukan maka waktu yang dibutuhkan akan semakin lama.

Tabel 1 Istilah-Istilah dalam Algoritma Genetika

No	Istilah	Keterangan
1	Gen	Gen merupakan nilai yang menyatakan satuan dasar yang membentuk satu kesatuan yang dinamakan kromosom.

No	Istilah	Keterangan
2	Kromosom	Kromosom merupakan gabungan dari gen-gen yang membentuk nilai tertentu yang menyatakan solusi yang mungkin dari suatu permasalahan.
3	Populasi	Populasi merupakan kumpulan dari individu yang diproses bersama dalam suatu siklus evolusi
4	<i>Fitness</i>	<i>Fitness</i> merupakan nilai dari seberapa baik individu yang didapatkan.
5	Seleksi	Seleksi adalah proses pemilihan individu yang baik untuk dijadikan induk.
6	<i>Crossover</i>	<i>Crossover</i> merupakan proses pertukaran atau kawin silang gen-gen dari dua induk tertentu.
7	Mutasi	Mutasi merupakan proses pergantian salah satu gen yang terpilih dengan nilai tertentu.
8	Generasi	Generasi merupakan urutan iterasi dimana beberapa kromosom bergabung.
9	<i>Offspring</i>	<i>Offspring</i> merupakan kromosom baru yang dihasilkan (kromosom anak)

Struktur Umum Algoritma Genetika

Pada Algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin (populasi). Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas suatu kromosom dalam populasi tersebut.

Generasi berikutnya dikenal dengan istilah anak (*offspring*), terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*). *Offspring* dibentuk dari dua kromosom dengan operator penyilangan (*crossover*). Selain dengan *crossover*, kromosom dapat dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik (10).

Teknik Penyandian

Teknik penyandian ini meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk: *string* bit, pohon, *array* bilangan real, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika (10).

Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada (10).

Nilai Fitness

Suatu Individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi akan bertahan hidup. Sedangkan individu bernilai *fitness* rendah akan mati (11).

Untuk menemukan fungsi *fitness* yang tepat untuk suatu masalah, yang harus diperhatikan adalah fungsi objektif (*objective function*). Pada kasus optimasi, dikenal dua masalah yaitu: masalah maksimasi atau masalah minimasi. Maksimasi artinya mencari nilai maksimal dari sesuatu. Sedangkan minimasi artinya mencari nilai minimal dari sesuatu (12).

Menurut Haupt dan Haupt (12), Jika tujuannya adalah memaksimalkan sebuah fungsi, maka fungsi tujuan *fitness* yang digunakan adalah fungsi itu sendiri, misalkan masalahnya memaksimalkan fungsi h , maka formula fungsi *fitness* yang bisa digunakan adalah $f = h$ (dimana f adalah fungsi *fitness*).

Tetapi jika tujuannya adalah meminimalkan fungsi h , maka fungsi h tidak bisa digunakan secara langsung. Hal ini, disebabkan Algoritma Genetika menggunakan suatu aturan bahwa individu yang memiliki nilai *fitness* lebih tinggi memiliki kemampuan bertahan hidup lebih tinggi dari pada individu yang bernilai *fitness* rendah. Oleh karena itu, fungsi *fitness* untuk masalah minimasi adalah $f = 1/h$. Artinya, semakin kecil nilai h semakin besar nilai f . Tetapi fungsi *fitness* ini akan bermasalah jika h bisa bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasi masalah tersebut, fungsi *fitness* perlu dimodifikasi sedikit menjadi:

$$f = \frac{1}{(h+a)}$$

Dimana a adalah bilangan yang dianggap sangat kecil dan disesuaikan dengan masalah yang akan diselesaikan.

Seleksi

Seleksi ini melakukan pemilihan dua buah kromosom untuk diadakan sebagai induk (*parent*) (11). Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Ada beberapa metode seleksi dari induk yaitu (10):

1. Rank based fitness assignment
2. Roulette wheel selection
3. Stochastic universal sampling
4. Local Selection
5. Truncation selection
6. Tournament selection

Langkah pertama pada tahap seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut (10).

Metode Seleksi Roda Roulette

Metode seleksi ini merupakan metode seleksi yang paling sederhana, dan sering juga dikenal dengan nama *stochastic sampling with replacement* (10). Sesuai dengan namanya metode ini menirukan permainan *roulette wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda roulette secara proporsional sesuai dengan nilai *fitness*-nya. Kromosom yang memiliki nilai *fitness* lebih besar menempati potongan lingkaran yang lebih besar dibandingkan kromosom dengan nilai *fitness* yang lebih kecil (11).

Crossover

Crossover atau pindah silang ini merupakan komponen yang penting dalam AG. Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom (11). *Crossover* pada dua buah induk akan menghasilkan dua buah kromosom baru yang sering disebut kromosom anak (*offspring*), sehingga jumlah populasi akan bertambah dua kali dari jumlah populasi awal (12).

Kemungkinan dari beberapa individu yang akan mengalami *crossover* pada suatu populasi disebut dengan peluang *crossover* (P_c). Semakin besar nilai P_c semakin banyak individu dalam populasi yang akan mengalami *crossover*. Misalkan nilai P_c yang ditentukan adalah 0,2 dan ukuran populasi adalah 100, maka diharapkan terdapat 20 kromosom dari 100 kromosom pada populasi tersebut akan mengalami *crossover* (12).

Menurut Putra (12) metode *crossover* yang paling sering digunakan adalah *one-point crossover* (*crossover* satu titik). Metode ini melakukan pemotongan pada dua induk

menjadi dua bagian di titik yang sama. Titik potong pada metode ini dipilih secara acak. Potongan kromosom pada dua induk kemudian saling ditukarkan.

Induk1

1	1	0	0	0	0
---	---	---	---	---	---

Induk2

0	0	1	1	1	1
---	---	---	---	---	---

Titik potong

Anak1

1	1	1	1	1	1
---	---	---	---	---	---

Gambar 1. *One-point Crossover*

Anak2

0	0	0	0	0	0
---	---	---	---	---	---

Mutasi

Setelah mengalami proses *crossover*. Pada *offspring* dapat dilakukan mutasi. Haupt dan Haupt (12) menjelaskan mutasi merupakan proses mengubah secara acak nilai satu atau beberapa gen dalam satu kromosom. Variabel *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil, dengan probabilitas rendah.

Peluang mutasi (P_m) didefinisikan sebagai persentasi mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi (10). Kusumadewi menyebutkan jika P_m terlalu kecil, akan ada banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi jika P_m terlalu besar, maka akan ada terlalu banyak gangguan acak sehingga anak akan kehilangan kemiripan dengan induknya, dan algoritma akan kehilangan kemampuan untuk belajar dari histori pencarian.

Proses mutasi dalam pengkodean nilai dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan memilih sembarang posisi gen pada kromosom, nilai yang ada tersebut kemudian dirubah dengan suatu nilai tertentu yang diambil secara acak.

Contoh:

Kromosom sebelum mutasi: 29637

Kromosom sesudah mutasi: 21647

Keuntungan Algoritma Genetika

Untuk masalah optimasi, algoritma genetika dapat digunakan untuk optimasi numerik dan optimasi kombinatorial seperti *Traveling Salesman Problem (TSP)*, Perancangan *Integrated Circuit (IC)*, *Job Shop Scheduling*, serta optimasi video dan suara. Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari implementasi dan kemampuannya untuk menemukan solusi yang bagus secara cepat untuk masalah-masalah berdimensi tinggi. Algoritma genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut (11):

1. Ruang masalah yang besar, kompleks, dan sulit dipahami,
2. Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah kedalam ruang pencarian yang lebih sempit,
3. Tidak tersedianya analisis matematika yang memadai,
4. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi,
5. Solusi yang diharapkan tidak harus optimal, tetapi cukup bagus atau bisa diterima,
6. Terdapat batasan waktu, misalnya dalam sistem waktu nyata (*real time system*).

METODE

Tahapan pertama dari penelitian ini adalah melakukan pendefinisian masalah yang ingin diselesaikan. Setelah mendefinisikan masalah yang ingin dipecahkan langkah berikutnya adalah melakukan pengumpulan data (13) untuk mendukung penyelesaian permasalahan yang dihadapi. Setelah data yang diperlukan terkumpul, data dianalisis sebagai dasar dalam pembuatan aplikasi. Tahapan pembuatan aplikasi terdiri dari perancangan *database*, antarmuka, dan pembuatan kode program. Tahapan berikutnya adalah melakukan input data *sample* untuk menguji sistem. Jika sistem menghasilkan luaran yang tidak sesuai dengan harapan, maka

akan dilakukan kembali analisis data. Jika luaran telah sesuai dengan harapan maka tahapan penelitian telah selesai.

HASIL DAN PEMBAHASAN

Penyelesaian permasalahan pada penelitian ini akan diselesaikan dengan beberapa tahapan yaitu dengan tahapan sebagai berikut

Analisis Kebutuhan

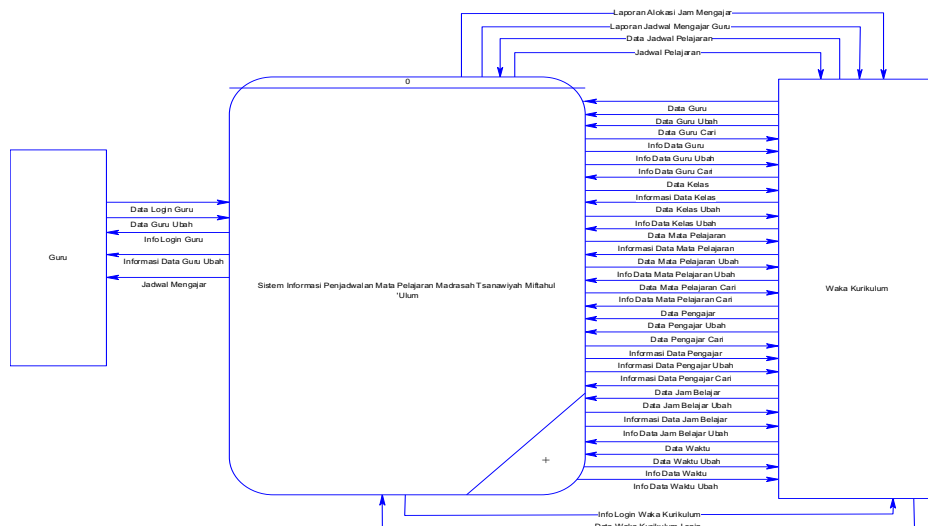
Analisis kebutuhan (14) dilakukan untuk mengetahui kebutuhan-kebutuhan dari Sistem Informasi Penjadwalan Pelajaran di Madrasah Tsanawiyah Miftahul 'Ulum Denpasar, yaitu antara lain:

- a) Proses pengelolaan data guru, data kelas, data mata pelajaran, data jam pelajaran dan data pengajar.
- b) Proses penjadwalan pelajaran dan laporan alokasi jam mengajar.

Perancangan Sistem

Perancangan Sistem Informasi Penjadwalan Pelajaran pada Miftahul 'Ulum Denpasar berikut menggunakan *data flow diagram (DFD)* (15) (16) (17). DFD digunakan untuk menggambarkan proses yang disediakan oleh sistem beserta pihak yang terlibat. Perancangan menggunakan DFD dimulai dari pembuatan diagram konteks untuk mengetahui pihak-pihak/kesatuan luar yang terlibat dalam sistem yang dibangun (gambar 1). Pada diagram ini terdapat satu proses dan 2 *external entities*, yaitu Waka Kurikulum selaku administrator dan guru. Perancangan sistem dilanjutkan dengan membuat DFD level 0 untuk mengetahui proses yang disediakan oleh sistem beserta keperluan penyimpanan datanya (gambar 2).

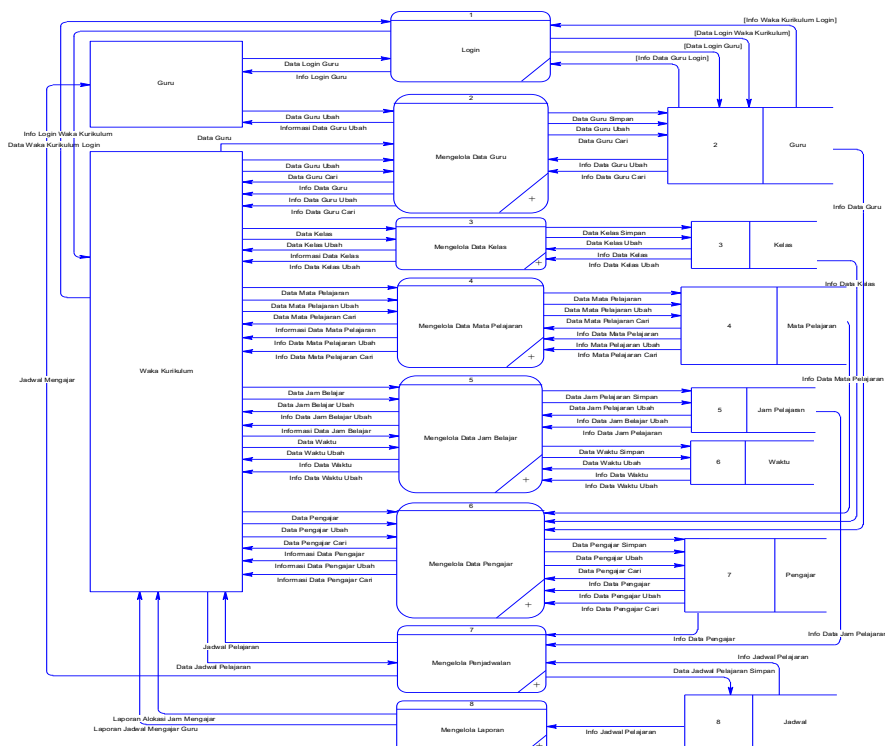
Pada DFD level 0 juga digambarkan keterlibatan entitas pada proses yang ada pada sistem. Rancangan basis data digunakan untuk mengetahui kebutuhan data yang disimpan untuk mendukung berjalannya sistem. Gambar 3 merupakan rancangan basis data dari sistem.



Gambar 2. Konteks Diagram

Konteks Diagram pada Gambar 2 menunjukkan keterlibatan entitas aliran data yaitu dari entitas Guru dan dari entitas Waka Kurikulum. Karena sistem yang dibangun hanya memerlukan data dari kedua entitas tersebut. Aliran data bergerak dari entitas guru menuju system dan sebaliknya, selain

itu aliran data juga bergerak dari entitas waka kurikulum ke sistem dan sebaliknya. Hal ini menyebabkan ketersediaan data menjadi terkendali dan sesuai dengan kebutuhan sistem.

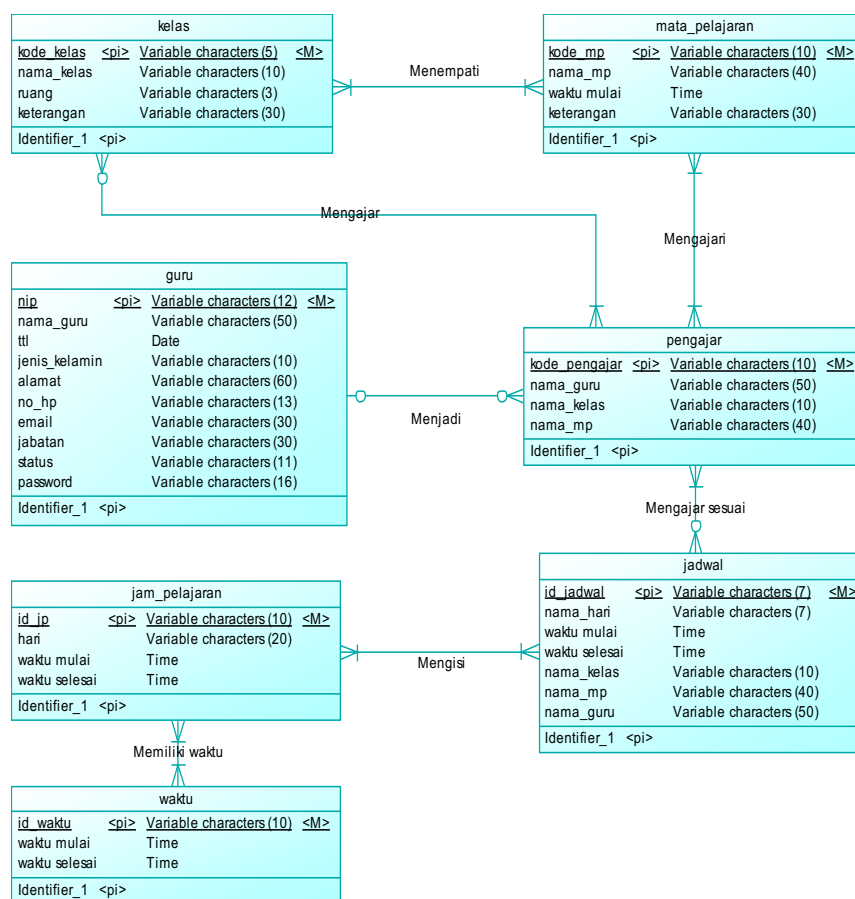


Gambar 3. DFD Level 0

Gambar 3. Menunjukkan breakdown dari gambaran umum di konteks diagram, pada Gambar 3 ini diperlihatkan banyaknya proses yang berjalan pada sistem penjadwalan dengan optimasi algoritma genetika. Terlihat pada DFD Level 0 terdapat 8 proses yang mendukung jalannya sistem penjadwalan ini. Selain 8 proses sistem ini juga terdapat 8 data store yang berfungsi sebagai penampung data yang ada pada sistem penjadwalan dengan optimasi algoritma genetika.

Selain menggunakan DFD dalam proses perancangan sistem, pada penelitian ini juga menggunakan *Conceptual Data Model (CDM)* (18) sebagai rancangan awal

dari database yang akan mendukung sistem penjadwalan dengan optimasi algoritma genetika ini. Adapun CDM di Gambar 4, memperlihatkan ada adanya 7 tabel yang saling berhubungan satu dengan yang lainnya. Relasi yang terbentuk dari CDM tersebut berjumlah 7 relasi, dengan 6 relasi many to many dan 1 relasi many to one. Relasi many to one terjadi pada relasi antara table guru dengan table pengajar, disana guru bisa melakukan berelasi ke pengajar dengan konsep 1 guru bisa mengajar beberapa mata pelajaran, dan beberapa mata pelajaran bisa di ajarkan oleh 1 guru.



Gambar 4. CDM Sistem Penjadwalan Algoritma Genetika

Tampilan Sistem

Tampilan sistem dikelompokkan menjadi dua halaman, halaman bagi administrator dan halaman bagi guru. Halaman administrator diperuntukkan bagi

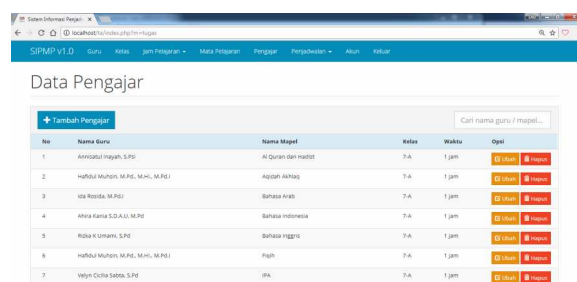
Waka Kurikulum untuk mengelola data guru, kelas, mata pelajaran, jam belajar dan tugas pengajar. Halaman administrator dapat dilihat pada gambar 5 sampai dengan gambar 9. Sedangkan halaman guru berisi

informasi jadwal mengajar. Halaman guru dapat dilihat pada gambar gambar 9



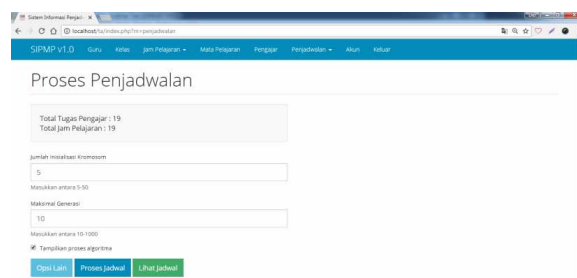
Gambar 5. Halaman Login

Halaman *Login* digunakan oleh *user* untuk masuk ke halaman menu utama. Halaman *login* utama di bedakan atas 2 *level* yaitu Waka Kurikulum sebagai *Admin* dan Guru.



Gambar 6. Data Pengajar

Tampilan pada Gambar 5 memperlihatkan bahwa *Administrator* dapat menambah data guru dengan mengklik tombol tambah pengajar dan mencari data guru di kolom pencarian di sisi kanan atas.



Gambar 7. Proses Penjadwalan

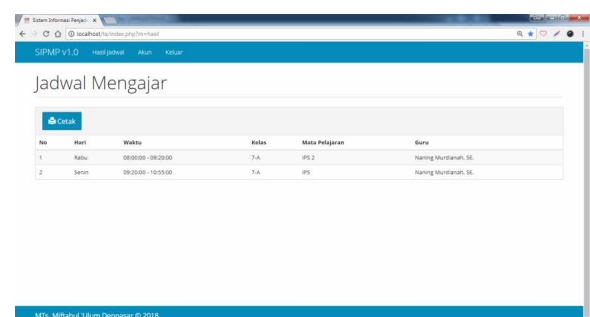
Pada gambar 6 merupakan halaman proses penjadwalan. *Admin* dapat membuat jadwal baru dengan operator genetika. Operator genetika yang diinputkan ke sistem meliputi jumlah kromosom, maksimal

generasi, persentase kawin silang (*crossover*) dan persentase mutasi gen



Gambar 8. Laporan Jadwal Mengajar Guru

Gambar 8 memperlihatkan hasil yang didapatkan setelah melakukan proses penjadwalan dengan memasukkan parameter dari algoritma genetika. Tentu saja hal ini akan bisa memeberikan hasil yang baik apabila sebelum melakukan proses penjadwalan dilakukan setting jam pelajaran terlebih dahulu.



Gambar 9. Jadwal Mengajar

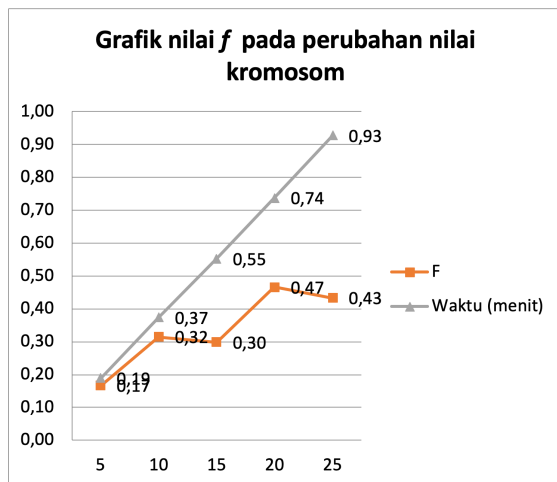
Gambar 9 menunjukkan tampilan jadwal mengajar yang dapat dilihat oleh user guru. Jadwal mengajar ini akan terlihat setelah proses penjadwalan dijalankan pada sistem penjadwala dengan optimasi algoritma genetika.

Pengujian Sistem

Pada penelitian ini, uji kinerja yang dilakukan untuk mengukur bagaimana pengaruh jumlah kromosom, jumlah generasi, nilai probabilitas *crossover* (*Pc*) serta probabilitas mutasi (*Pm*). Pengujian dilakukan sebanyak 10 kali sehingga mendapatkan rata-rata nilai *fitness* tertinggi.

Dalam pengujian pertama, ditentukan nilai maksimal generasi 100, nilai *Pc* adalah 0,1dan nilai *Pm* adalah 0.01. Setelah diberi variabel nilai kromosom yang berbeda,

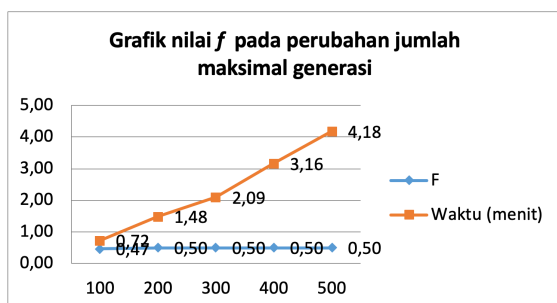
menghasilkan grafik seperti pada gambar 10 berikut ini.



Gambar 10. Grafik hasil pengujian variabel nilai kromosom

Dari grafik diatas, dapat disimpulkan bahwa jumlah kromosom tidak mempengaruhi nilai *fitness* secara signifikan, tetapi mempengaruhi waktu proses penjadwalan. Semakin banyak jumlah kromosom, maka semakin lama pula waktu yang dibutuhkan

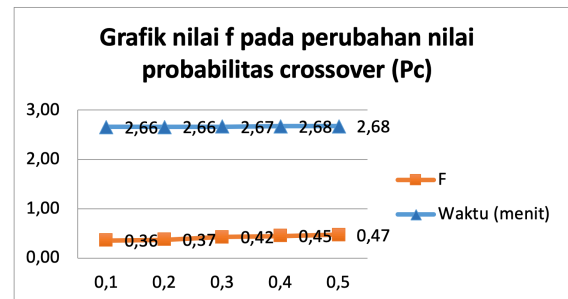
Selanjutnya Pada pengujian kedua, ditentukan nilai kromosom terbaik dari gambar 4.27, nilai P_c adalah 0,1 dan nilai P_m adalah 0.01. Setelah diberi variabel nilai generasi yang berbeda, menghasilkan grafik seperti pada gambar 11.



Gambar 11. Grafik hasil pengujian variabel jumlah generasi

Dari pengujian diatas, dapat disimpulkan bahwa nilai maksimal generasi hanya mempengaruhi waktu proses dan tidak mempengaruhi nilai *fitness* secara signifikan

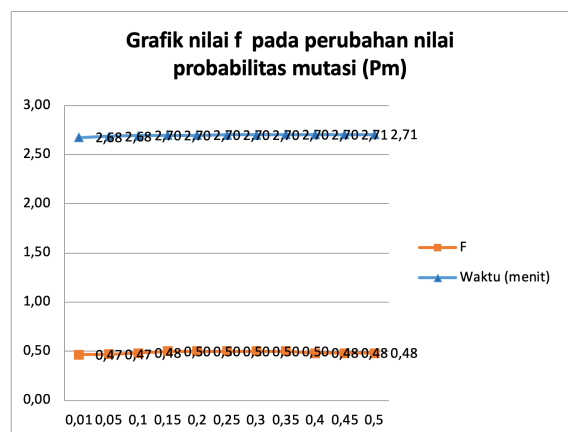
Pada pengujian ketiga, ditentukan nilai kromosom adalah 20, nilai maksimal generasi 200 (dipilih karena waktu tercepat adalah pada nilai 200) serta nilai P_m adalah 0,01. Setelah diberi variabel nilai P_m yang berbeda, menghasilkan grafik seperti pada gambar 12



Gambar 12. Grafik hasil pengujian variabel nilai P_c

Dari pengujian diatas, dapat disimpulkan bahwa nilai probabilitas *crossover* (P_c) dapat mempengaruhi nilai *fitness*, dan tidak mempengaruhi waktu secara signifikan.

Selanjutnya pada pengujian keempat, ditentukan nilai kromosom sebesar 20, nilai maksimal generasi sebesar 200 dan nilai P_c adalah sebesar 0.5. Setelah diberi variabel nilai P_m mulai dari 0,01 hingga 0,5 dengan jarak interval 0,05 menghasilkan grafik seperti pada gambar 13.



Gambar 13. Grafik hasil pengujian variabel nilai P_m

Dari pengujian diatas, dapat disimpulkan bahwa nilai probabilitas mutasi (Pm) tidak mempengaruhi waktu proses secara signifikan. Nilai *fitness* terbaik dicapai pada nilai Pm minimal 0,15 dan nilai Pm maksimal 0,35.

Dari keempat pengujian diatas, didapatkan kombinasi operator genetika terbaik dengan nilai *fitness* 0,5 yaitu pada nilai kromosom sebesar 20 buah, jumlah generasi sebesar 200, nilai probabilitas *crossover* (Pc) sebesar 0,5 dan nilai probabilitas mutasi (Pm) minimal sebesar 0,15 dan maksimal sebesar 0,35. Pada nilai probabilitas mutasi (Pm) sebesar 0,15 dibutuhkan waktu selama 91,22 detik, sedangkan pada nilai Pm sebesar 0,35 dibutuhkan waktu selama 91,65 detik.

SIMPULAN

Berdasarkan hasil implementasi dari penelitian yang penulis buat tentang Sistem Penjadwalan Mata Pelajaran pada Madrasah Tsanawiyah Miftahul 'Ulum Denpasar menggunakan Algoritma Genetika, maka penulis menarik kesimpulan sebagai berikut:

1. Penggunaan metode optimasi algoritma genetika dalam penelitian ini hanya menggunakan 2 *hard constraint* atau batasan, yaitu bentrok jadwal guru yang bersamaan dalam satu waktu dan bentrok kelas yang sama dalam satu waktu.
2. Pengujian sistem dilakukan dengan memberi beban jadwal sejumlah 5 kelas yang diatur ke dalam 19 slot jam mulai hari senin hingga jumat (panjang kromosom terhitung 95 gen). Dari beberapa kombinasi operator algoritma genetika yang ada, solusi dengan nilai *fitness* terbaik dicapai dengan jumlah kromosom sebesar 20 buah, jumlah generasi 200, nilai probabilitas *crossover* sebesar 0,5 dan nilai probabilitas mutasi (Pm) berkisar mulai dari 0,15 hingga 0,35. Pada nilai minimal Pm 0,15, waktu yang dibutuhkan untuk memproses penjadwalan adalah selama 91,22 detik, sedangkan pada nilai maksimal Pm sebesar 0,35, waktu yang dibutuhkan untuk memproses penjadwalan adalah selama 91,65 detik.

3. Dari pengujian beberapa nilai operator genetika, penulis menyimpulkan bahwa nilai *fitness* dipengaruhi oleh jumlah kromosom dan nilai probabilitas *crossover*, sedangkan waktu proses dipengaruhi oleh jumlah maksimal generasi.

REFERENSI

- [1] Mawaddah NK, Mahmudy WF. Optimasi Penjadwalan Perawat Menggunakan Algoritma Genetika. Univ Brawijaya. 2006;
- [2] Desiana E. Performance Algoritma Genetika (GA) Pada Penjadwalan Mata Pelajaran. J Nas Inform dan Teknol Jar. 2016;
- [3] Susanto A. Sistem Informasi Manajemen. Bandung: Lingga Jaya; 2007.
- [4] García-Martínez C, Rodríguez FJ, Lozano M. Genetic algorithms. In: Handbook of Heuristics. 2018.
- [5] Kubat M, Kubat M. The Genetic Algorithm. In: An Introduction to Machine Learning. 2017.
- [6] Cao YJ, Wu QH. Teaching Genetic Algorithm Using Matlab. Int J Electr Eng Educ. 2013;
- [7] Wati DAR, Rochman YA. Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO). J Rekayasa Sist Ind. 2013;
- [8] Setemen K. Implementasi Algoritma Genetika Dalam Pengembangan Sistem Aplikasi Penjadwalan Kuliah. J IKA. 2010;
- [9] KraleV V. A Genetic and Memetic Algorithm for Solving The University Course Timetable Problem. Int J "Information Theor Appl. 2009;
- [10] Kusumadewi S. Artificial Intelligence. Yogyakarta: Graha Ilmu; 2003.
- [11] Suyanto. Algoritma Genetika dalam Matlab. Yogyakarta: Penerbit Andi; 2005.
- [12] Putra DMDU. Penerapan Algoritma Genetika untuk Menyelesaikan Permasalahan Perawat dengan Fuzzy Fitness Function. 2012;
- [13] Martono N. Metode Penelitian

- [14] Kuantitatif Analisis Isi dan Analisis Data Sekunder. Ed Revisi 2. 2014; Iswara IBAI, Saputra IGA. Aplikasi Incosys Sebagai Alternatif System Pengaduan Online. J Nas Pendidik Tek Inform. 2017;6(3):316–27.
- [15] Wieringa RJ. Data Flow Diagrams. In: Design Methods for Reactive Systems. 2007.
- [16] Ibrahim R, Yen SY. Formalization of the Data Flow Diagram Rules for Consistency Check. Int J Softw Eng Appl. 2010;
- [17] Mulawarman MF. Sistem Informasi Penjualan Pembelian Dan Persediaan Barang Pada Toko Yunika Berbasis Desktop. Issn. 2015;1(Strata 1).
- [18] Sudarmaji S, Prasetyo KA. SISTEM INFORMASI PERKULIAHAN ONLINE PADA FAKULTAS ILMU KOMPUTER UNIVERSITAS MUHAMMADIYAH METRO LAMPUNG. J Nas Pendidik Tek Inform [Internet]. 2019 Jun 10 [cited 2019 Jul 29];8(1):12. Available from: <https://ejournal.undiksha.ac.id/index.php/janapati/article/view/14010>