

ABSTRAK

Penjadwalan mata kuliah merupakan masalah yang sangat kompleks yang dihadapi oleh pihak jurusan (sekretariat) dalam sebuah universitas. Penjadwalan mata kuliah ini terkait dengan ketersediaan dosen, mahasiswa, ruangan kuliah, hari, jam, dan mata kuliah yang ditawarkan. Sementara hasil yang diharapkan adalah adanya jadwal mata kuliah yang sesuai, baik bagi dosen maupun mahasiswa dengan waktu kuliah yang efektif, sehingga tidak terjadi tabrakan baik antara jadwal mata kuliah yang satu dengan jadwal mata kuliah yang lain, maupun jadwal bagi dosen sebagai pengajar mata kuliah.

Persoalan penjadwalan mata kuliah ini dapat diselesaikan menggunakan Algoritma Genetika. Algoritma Genetika merupakan sebuah solusi alternatif untuk menyelesaikan masalah optimasi yang berdasarkan pada mekanisme seleksi alami dan sifat genetika. Dari data yang ada, berupa data dosen, data ruang, data hari, data jam, dan data mata kuliah dapat diarahkan untuk pembentukan model kromosom. Sebuah kromosom dibangun dari komponen (variabel) mata kuliah, hari, ruang dan jam, dimana hari, ruang, dan jam terlebih dahulu diarahkan menjadi sebuah kode (sebuah kode menyatakan Hari tertentu, Ruang tertentu, dan Jam tertentu). Panjang sebuah kromosom (yang menyatakan jumlah gen) adalah hasil kali antara $Jumlah_Hari * Jumlah_Ruang * Jumlah_Jam$.

Proses Algoritma Genetika dimulai dengan membangkitkan kromosom-kromosom sebagai populasi awal. Setiap kromosom memiliki tingkat kesesuaian berdasarkan nilai *fitness*-nya. Nilai *fitness* yang terbaik memiliki kesempatan untuk melakukan proses rekombinasi (sebagai induk) yaitu perkawinan silang dan mutasi. Dalam penjadwalan mata kuliah, kromosom akan dibangkitkan secara random menggunakan pengkodean permutasi, perkawinan silang (*Crossover*) menggunakan metode *Partial Mapped Crossover* (PMX) dan mutasi menggunakan metode mutasi pertukaran, sedangkan nilai *fitness* didapat dari jumlah pelanggaran terhadap aturan-aturan yang telah ditetapkan. Hasil rekombinasi (keturunan) akan di-evaluasi, apakah nilai *fitness* = 0 atau tidak. Jika nilai *fitness* = 0 atau iterasi = 1000 (jumlah maksimal yang telah ditetapkan untuk melakukan proses Algoritma Genetika), maka proses akan berhenti, tetapi jika nilai *fitness* \diamond 0 dan iterasi \diamond 1000, maka hasil rekombinasi akan diseleksi untuk membentuk populasi yang baru.

Proses Algoritma Genetika akan berhenti jika memiliki nilai *fitness* = 0 atau iterasi = 1000, proses ini akan menghasilkan nilai *fitness* = 0 atau nilai *fitness* \diamond 0. Nilai *fitness* = 0 artinya tidak ada lagi pelanggaran terhadap aturan-aturan yang telah ditetapkan atau dengan kata lain tidak ada jadwal mata kuliah dalam satu paket (semester) yang tabrakan, demikian juga dengan jadwal dosen sebagai pengajar. Akan tetapi, jika nilai *fitness* \diamond 0, maka masih ada jadwal mata kuliah yang tabrakan.

ABSTRACTS

Class scheduling is a complex problem for the faculty administration in a university. Class scheduling is closely related to the availability of lecturers, students, classrooms, class days and time, and subjects offered. The role of Class Scheduling is to generate schedule for all the subjects offered, without any collision between the subjects.

This Class Scheduling problem is resolved with genetic algorithm. Genetic algorithm is an alternative solution to resolve the optimization problem based on natural selection mechanism and genetic nature. From the available data, that is the data of lecturers, classrooms, class days and time, and subjects offered, a chromosome model is generated. A chromosome is built from subjects, days and time, and classrooms variables. Each variable is designated with a code. The length of a chromosome is the result of the variables combinations.

The genetic algorithm process begins with the generation of chromosomes as early population. Each chromosome has a level of fitness according to its fitness value. The best fitness value has the opportunity for a recombining process (as parent), that is a crossover and mutation. In class scheduling, chromosomes are raised randomly using permutation coding, crossover using partial mapped crossover (PMX) method), and mutation using displacement mutation method, whereas the fitness value is acquired by the number of violations of the rules established. The result of the recombination will be evaluated, to see whether the fitness value is equal to zero ($= 0$) or not. If it is zero, or the iteration is equal to 1000 (the maximum number has been established to perform the genetic algorithm process) than the process will end, but if it is not, than the recombination result will be selected to generate a new population.

The Genetic Algorithm process will end if the fitness value acquired is equal to zero ($= 0$) or the iteration is equal to one thousand ($= 1000$). This process will generate fitness value $= 0$, or fitness value is more or less than zero ($< > 0$). Fitness value $= 0$ means that there is no more violations to the rules established, or in other words there is no class schedule in one package (term) that collide with other classes. If the fitness value is more or less than zero ($< > 0$), it means that there are classes that collide.