

BAB III

LANDASAN TEORI

A. Penjadwalan

Penjadwalan merupakan proses pengalokasian sumber daya seperti waktu, tenaga kerja, dan mesin secara efisien untuk menyelesaikan serangkaian aktivitas dalam urutan tertentu guna mencapai tujuan tertentu[22]. Dalam konteks produksi maupun pelayanan, penjadwalan sangat penting untuk memastikan bahwa proses berjalan lancar tanpa terjadi penumpukan pekerjaan atau keterlambatan. Menurut Krajewski dan Ritzman penjadwalan adalah pengalokasian sumber daya dari waktu ke waktu untuk menunjang pelaksanaan dan penyelesaian suatu aktivitas tertentu[23]. Sementara itu, Sirat dan Ginting menjelaskan penjadwalan sebagai alat pengambilan keputusan yang melibatkan urutan pekerjaan yang harus diselesaikan pada beberapa mesin atau sumber daya[23].

B. Penjadwalan Mata Pelajaran di Sekolah Dasar

Kegiatan penjadwalan ini sering terjadi di berbagai sektor mulai dari industri untuk bagian produksi hingga pendidikan untuk menyusun jadwal mata pelajaran atau yang bisa juga disebut penjadwalan mata pelajaran. Penjadwalan mata pelajaran merupakan suatu proses penyusunan jadwal kegiatan pembelajaran di lingkungan sekolah yang melibatkan alokasi waktu, ruang dan sumber daya pendidikan seperti guru, siswa serta mata pelajaran ke dalam slot waktu tertentu dalam satuan minggu atau semester.

Pada sekolah dasar, proses penjadwalan harus mempertimbangkan keterbatasan jam mengajar guru, kuota jam pelajaran yang harus di penuhi setiap mata pelajaran dan aturan dari kurikulum. Penjadwalan yang baik akan menjamin bahwa tidak terjadi tumpang tindih dalam penggunaan ruang kelas, waktu mengajar guru serta pembagian jam pelajaran untuk siswa. Namun, penyusunan jadwal pelajaran tidak mudah, terutama apabila dilakukan secara manual atau konvensional[24].

C. Optimasi Penjadwalan

Masalah penjadwalan pelajaran di sekolah dasar merupakan salah satu contoh dari masalah optimasi kombinatorial. Optimasi kombinatorial muncul ketika terdapat sejumlah besar kemungkinan solusi (kompensasi antara *constraint* atau batasan dan variabel) dan pencarian solusi optimal harus mempertimbangkan berbagai batasan (*hard* dan *soft constraint*). Dalam konteks sekolah, *hard constraint* bisa berupa guru tidak dapat mengajar dua kelas sekaligus, ruangan hanya dapat digunakan satu kelas dalam satu waktu, dan sebagainya. *Soft constraint* meliputi pemerataan jam mengajar, preferensi jam tertentu untuk guru tertentu dan sebagainya.

D. Algoritma Genetika

Masalah optimasi kombinatorial pada penjadwalan mata pelajaran di sekolah dasar ini merupakan salah satu contoh kasus yang dapat diselesaikan menggunakan pendekatan komputasi evolusi atau komputasi dengan inspirasi dari proses evolusi alami seperti seleksi, mutasi dan lain sebagainya. Salah satu metode yang sudah terbukti cocok untuk masalah optimasi susunan jadwal ini adalah algoritma genetika.

Algoritma Genetika merupakan metode pencarian heuristik berdasarkan prinsip evolusi dan seleksi alam. Algoritma genetik bekerja dengan merepresentasikan solusi (jadwal pelajaran) sebagai kromosom yang terdiri dari gen-gen (slot-slot waktu dan mata pelajaran). Populasi solusi secara berulang mengalami seleksi, *crossover* (pertukaran segmen), dan mutasi (perubahan acak) untuk menghasilkan solusi baru. Proses ini berlangsung hingga ditemukan solusi optimal atau mencapai kriteria penghentian tertentu. Komponen algoritma genetika terdiri dari kromosom, *fitness value*, seleksi, *crossover*, mutasi dan populasi[25].

1. Kromosom (representasi solusi)

Dalam konteks algoritma genetika, kromosom merupakan representasi dari solusi potensial terhadap permasalahan optimasi yang sedang diselesaikan. Unit terkecil dari sebuah kromosom atau solusi biasa disebut sebagai gen. Kromosom ini umumnya dikodekan dalam

bentuk *string* simbol seperti bit biner dan dalam hal ini gen merupakan setiap bit biner tersebut[26]. Namun, dalam beberapa kasus tertentu dapat dikodekan dalam bentuk lain atau bahkan menggunakan data aslinya atau *real value* hanya dengan mengubah struktur datanya saja tergantung struktur data masalah dan solusi optimasi yang sedang diselesaikan. Hal ini dilakukan untuk membuat representasi dari sebuah solusi ke dalam formula tertentu sehingga dapat mempermudah dalam melakukan beberapa operasi selanjutnya yang terjadi dalam algoritma genetik dalam rangka proses pencarian solusi optimal.

2. *Fitness Value*

Pada setiap iterasi, setiap kromosom akan di evaluasi menggunakan suatu fungsi evaluasi yang biasa disebut *fitness function*. Fungsi ini bertugas untuk mengukur seberapa baik kualitas solusi yang direpresentasikan oleh kromosom tersebut dalam menyelesaikan permasalahan yang sedang di selesaikan. Hasil dari evaluasi ini disebut *fitness value* dan hasil tersebut digunakan untuk menentukan tingkat kelayakan kromosom dalam berpartisipasi pada proses iterasi berikutnya[27].

Terdapat beberapa metode untuk mencari atau menghitung nilai *fitness* ini. Setiap masalah optimasi yang ingin diselesaikan memiliki rumus untuk mencari nilai *fitness* yang sedikit berbeda, namun yang paling umum digunakan adalah rumus berikut.

$$f = \frac{1}{(\text{fungsi objektif})}$$

Rumus di atas merupakan salah satu rumus yang paling sering digunakan untuk mencari atau menghitung nilai *fitness* dari sebuah kromosom yang menandakan kualitas dari kromosom atau representasi dari solusi tersebut dengan batasan – batasan atau aturan – aturan yang diberikan. Fungsi objektif pada rumus tersebut merupakan bagian dari rumus lain yang akan dihitung terlebih dahulu. Fungsi objektif tersebut yang berbeda setiap kasus, contohnya adalah seperti rumus berikut.

$$\begin{cases} f = \frac{1}{SE} \\ SE = \sum (yi - oi)^2 \end{cases}$$

Rumus tersebut merupakan rumus yang digunakan untuk mencari nilai *fitness* untuk masalah prediksi harga produk *sillicon wafer* di negara China[28]. Perhitungan nilai *fitness* dilakukan dengan mencari terlebih dahulu fungsi objektif yang dihitung dengan rumus $SE = \sum (yi - oi)^2$ di mana simbol *yi* merupakan nilai yang diharapkan, sedangkan simbol *oi* merupakan nilai yang dihasilkan lalu dikuadratkan sehingga merepresentasikan nilai *error* dari kromosom atau solusi yang di evaluasi[28]. Lalu nilai satu dibagi dengan hasil nilai *error* yang sudah dihitung dengan fungsi objektif tadi sehingga dapat diketahui nilai *fitness* dari solusi tersebut.

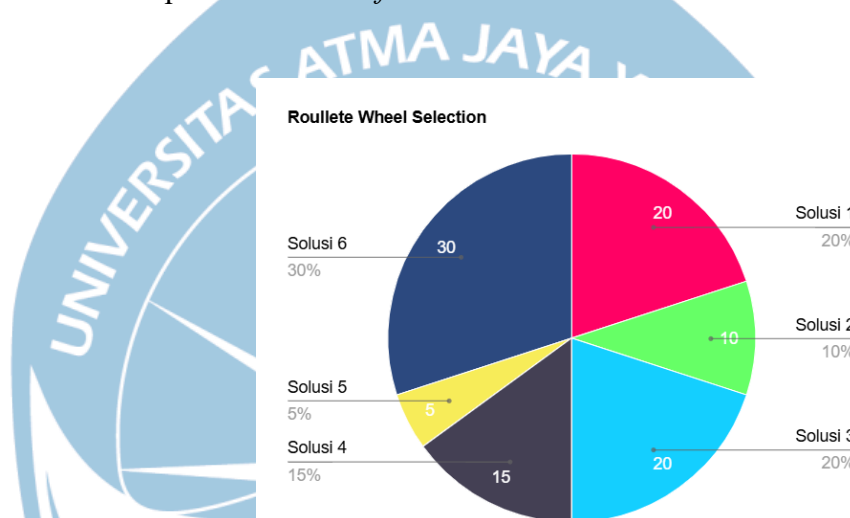
3. Seleksi

Seleksi adalah proses memilih dua individu yang akan dijadikan *parent* atau orang tua dari proses selanjutnya, yaitu *crossover* atau pindah silang. Proses seleksi ini dilakukan berdasarkan hasil perhitungan *fitness value* yang didapatkan sebelumnya serta proses ini dilakukan pada setiap individu dengan memasukkannya pada rumus sesuai metode seleksi yang digunakan. Terdapat beberapa metode seleksi yang ada dalam algoritma genetika, yaitu *roullete wheel*, *elitism*, seleksi rangking (*rank selection*), *stochastic universal sampling*, seleksi lokal, seleksi turnamen, dan masih banyak lagi[27].

Salah satu metode seleksi yang paling sering digunakan adalah *roullete wheel*. Metode seleksi ini bekerja dengan cara menghitung probabilitas terpilihnya suatu representasi dari sebuah solusi atau yang bisa disebut juga kromosom berdasarkan nilai *fitness* yang dimiliki oleh kromosom tersebut[29]. Secara umum perhitungan tersebut dilakukan dengan menggunakan rumus berikut.

$$pi = \frac{fi}{\sum_{k=1}^n fk}$$

Berdasarkan rumus tersebut, probabilitas suatu solusi atau kromosom dipilih dihitung dengan cara nilai *fitness* dari solusi tersebut yang direpresentasikan dengan simbol f_i dalam rumus dibagi dengan total nilai *fitness* seluruh kromosom atau solusi pada populasi yang direpresentasikan dengan rumus $\sum_{k=1}^n f_k$ di mana simbol n merupakan jumlah kromosom atau solusi pada populasi, sedangkan simbol f_k merupakan total nilai *fitness* keseluruhan.



Gambar 3.1 Visualisasi Metode Seleksi *Roulette Wheel*

Pada Gambar 3.1 merupakan visualisasi dari proses seleksi dengan metode *roulette wheel* yang bisa dianalogikan seperti memutar roda undian dengan perbedaan luas antar solusi yang dihitung menggunakan rumus yang menghasilkan probabilitas terpilihnya solusi tersebut. Dari contoh pada Gambar 3.1 tersebut terlihat bahwa solusi 6 memiliki probabilitas paling tinggi daripada solusi yang lain dan ini memberikan peluang terpilihnya solusi tersebut menjadi yang terbesar.

Salah satu metode seleksi yang sering digunakan lainnya adalah metode seleksi *elitism*. Metode seleksi ini akan menyimpan solusi terbaik berdasarkan dengan nilai *fitness* dari solusi tersebut sehingga mencegah hilangnya solusi terbaik tersebut ketika proses *crossover* dan mutasi dilakukan pada populasi[30]. Metode seleksi ini secara umum dapat direpresentasikan dalam rumus berikut.

$$E_i = \arg \text{top}_k(f(x)) \quad x \in P_i$$

Rumus tersebut merupakan salah satu contoh umum bagaimana proses seleksi *elitism* dilakukan. Untuk mengumpulkan kromosom atau solusi dengan nilai *fitness* terbaik atau yang bisa disebut sebagai *elit* dari sebuah populasi solusi yang direpresentasikan dalam rumus sebagai E_i yang artinya adalah *elit* pada generasi atau iterasi ke i , maka digunakan operator $\arg \text{top}_k(f(x))$ di mana simbol k merepresentasikan jumlah solusi terbaik yang ingin diambil dari populasi berdasarkan nilai dari simbol $f(x)$ yang merupakan representasi dari fungsi *fitness* di mana nilai dari simbol x merupakan anggota atau bagian dari populasi pada generasi atau iterasi ke i yang di representasikan sebagai simbol P_i pada rumus.

4. *Crossover* (pindah silang)

Setelah dilakukan seleksi maka proses selanjutnya adalah *crossover* atau pindah silang atau bisa juga disebut rekombinasi. Proses ini bertujuan untuk memberikan keanekaragaman solusi yang dihasilkan dari dua kromosom atau solusi pilihan dengan *fitness value* terbaik yang biasa disebut dengan *parent* dengan harapan menemukan solusi baru yang lebih baik. Proses *crossover* ini dapat dilakukan dengan berbagai cara tergantung metode yang digunakan. Beberapa *crossover* antara lain seperti *crossover* satu titik (*one point crossover*), *crossover* banyak titik (*n point crossover*), *arithmetic crossover* dan masih banyak lagi [27]. Peluang *crossover* ini di atur pada dengan metode seleksi tertentu, jadi setiap individu tidak selalu di kawinkan walaupun *fitness value* nya bagus dan atau terpilih dari hasil seleksi sebelumnya.

Salah satu metode *crossover* yang paling sering digunakan adalah *one point crossover*. Metode *crossover* ini dilakukan dengan cara memotong daftar data atau *array* sebuah data pada satu titik. Titik ini ditentukan secara acak dengan rumus berikut.

$$c = \text{randint}(1, n - 1)$$

Rumus tersebut akan menghasilkan angka bulat atau *integer* secara acak sebagai titik *crossover* atau titik potong yang direpresentasikan dengan simbol c dengan bantuan operator $\text{randint}(1, n - 1)$ yang memiliki arti angka yang dihasilkan adalah antara dari angka 1 sampai $n - 1$ di mana n adalah panjang indeks *array* dari kedua *parent* yang akan dilakukan operasi *crossover*. Representasi data kedua *parent* tersebut dapat dilihat pada ekspresi matematis berikut.

$$P^1 = [x_1^1, x_2^1, \dots, x_n^1]$$

$$P^2 = [x_1^2, x_2^2, \dots, x_n^2]$$

Pada rumus tersebut, simbol P^n merupakan representasi dari kromosom atau solusi ke n dengan isi datanya direpresentasikan dengan simbol x_i^j di mana i adalah gen atau unit terkecil dari sebuah solusi ke i , sedangkan simbol j merepresentasikan dari *parent* ke j . Kemudian, setelah titik potong ditentukan, maka selanjutnya adalah proses pemotongan kedua kromosom atau solusi yang akan dilakukan *crossover* pada indeks yang telah ditentukan tersebut. Proses ini dapat direpresentasikan dalam rumus berikut.

$$C_i^1 = \begin{cases} x_i^1, & i \leq c \\ x_i^2, & i > c \end{cases}, \quad C_i^2 = \begin{cases} x_i^2, & i \leq c \\ x_i^1, & i > c \end{cases}$$

Rumus tersebut merupakan representasi proses *crossover* yang terjadi pada setiap gen atau setiap unit terkecil dari sebuah kromosom atau solusi. Proses *crossover* ini akan menghasilkan dua kromosom atau solusi baru dengan kombinasi antara kedua *parent* ini yang dalam rumus direpresentasikan oleh simbol C_i^j di mana simbol i merepresentasikan gen ke i , sedangkan simbol j merepresentasikan *offspring* atau anak atau yang bisa disebut hasil *crossover* ke j .

Isi data yang ada pada setiap hasil *crossover* ini di atur oleh sebuah kondisi yang mengacu pada hasil penentuan titik potong di awal tadi yang direpresentasikan dengan dengan simbol c yang akan melakukan pengecekan jika indeks *array* kurang dari sama dengan nilai c , maka

masukkan data dari *parent* pertama jika pada *offspring* atau hasil *crossover* C_i^1 yang direpresentasikan dengan simbol x_i^1 , sedangkan jika indeks *array* lebih dari titik nilai c pada data *offspring* yang sama, maka masukkan data dari *parent* kedua yang direpresentasikan dengan simbol x_i^2 . Jika pada data hasil *crossover* C_i^2 atau *offspring* kedua, maka proses tadi tinggal dibalik saja.

5. Mutasi

Mutasi adalah proses mengganti salah satu nilai gen, yaitu unit terkecil dalam sebuah kromosom atau representasi solusi. Hal ini dilakukan dengan harapan memunculkan solusi yang lebih baik dari pada sebelumnya dengan perubahan lokal pada salah satu kromosom atau representasi solusi. Proses mutasi ini tentunya dikontrol dengan parameter peluang yang menentukan peluang terjadinya mutasi pada suatu kromosom, jika terlalu kecil maka tidak terlalu berpengaruh dan semakin lama untuk mencapai suatu konvergensi, sedangkan jika terlalu besar maka akan terjadi gangguan yang terlalu acak yang menyebabkan banyaknya potensi solusi yang terbuang[27]. Peluang ini dapat disebut sebagai *mutation rate* atau tingkat mutasi.

$$\langle x_1, \dots, x_n \rangle \rightarrow \langle x_1^t, \dots, x_n^t \rangle, \quad \text{where } x_i, x_i^t \in [L_i, U_i]$$

Rumus tersebut merupakan salah satu representasi bagaimana proses mutasi bekerja pada umumnya. Pada rumus tersebut, proses mutasi dilakukan pada kromosom atau representasi solusi yang didasari pada gen dengan *real valued representation* atau representasi dari nilai aslinya, bukan berupa *encoding* dari sebuah data. Inti dari rumus mutasi tersebut adalah mengganti anggota atau gen yang terdapat dilambangkan dengan vektor $\langle x_1, \dots, x_n \rangle$ yang berisi bilangan real bukan biner menjadi vektor $\langle x_1^t, \dots, x_n^t \rangle$ dengan nilai gen baru yang dilambangkan pada simbol x_i^t yang dihasilkan secara acak, namun masih tetap dalam batas yang diizinkan seperti pada simbol $[L_i, U_i]$ di mana simbol L_i merupakan batas bawah, sedangkan simbol U_i merupakan batas atas[31].

6. Populasi

Populasi merupakan kumpulan kromosom atau kumpulan solusi yang mungkin. Pada tahap awal algoritma genetika dijalankan, populasi awal di hasilkan secara acak maupun secara terstruktur sesuai dengan kriteria tertentu sebelum algoritma genetika memulai pencarian solusi. Ukuran populasi dipengaruhi oleh masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasi pada sistem[27].

7. Terminasi dan konvergensi

Proses pencarian solusi terbaik dalam algoritma genetika dilakukan secara berulang terus menerus. Oleh karena itu, perlu titik terminasi atau titik di mana proses tersebut berhenti. Titik ini bisa di definisikan di awal jika sistem algoritma genetika misalnya akan hanya berulang sebanyak 20 kali untuk mencari solusi terbaik. Setelah tercapai, hasil solusinya di evaluasi apakah sudah baik atau sesuai yang di inginkan atau belum. Jika sudah sesuai atau mendekati sesuai maka solusi tersebut dapat dikatakan sudah mencapai konvergensi.

E. Penerapan Sistem Informasi Berbasis Web

Pada umumnya, algoritma genetika dalam sistem penjadwalan dapat diterapkan dengan membuat sebuah sistem informasi berbasis web dengan fitur pembuatan atau penyusunan jadwal secara otomatis dengan bantuan algoritma genetika yang diterapkan dalam sistem. Sistem informasi berbasis web ini juga biasa disebut sebagai *website*. *Website* atau situs web merupakan kumpulan halaman digital yang saling terhubung dan dapat diakses melalui jaringan internet[32]. Setiap halaman dalam situs web yang dikenal sebagai laman web berisi berbagai jenis informasi dalam bentuk teks, gambar, animasi, suara maupun video yang disusun secara sistematis. Informasi tersebut disajikan dalam format digital dan dapat diakses oleh pengguna dari berbagai belahan dunia selama perangkat yang digunakan terhubung dengan jaringan internet.

Seiring dengan perkembangan teknologi informasi, *website* atau situs web tidak hanya menampilkan teks atau gambar statis, melainkan telah

berkembang menjadi media informasi yang bersifat multimedia. Artinya, informasi yang disajikan dalam situs web dapat di sampaikan melalui berbagai kombinasi media seperti teks, gambar, suara, animasi dan video yang di integrasikan sehingga memberikan pengalaman yang lebih interaktif dan menarik bagi pengguna sekaligus meningkatkan efektivitas penyampaian pesan atau informasi. Selain itu, situs web juga dapat menjadi sebuah alat *all in one* dengan tambahan teknologi di belakangnya yang akan memproses data yang di inginkan oleh pengguna sehingga semakin mempermudah pemrosesan data yang di perlukan. Hal ini membuat situs web dapat menjadi pilihan tepat dalam menerapkan algoritma genetik untuk menyelesaikan masalah penjadwalan mata pelajaran.

