

BAB 2

LANDASAN TEORI

2.1 Penjadwalan Mata Kuliah

Penjadwalan merupakan suatu kegiatan alokasi sumber daya dengan memiliki kendala (batasan) yang diberikan kepada suatu objek seperti di ruang-waktu, sedemikian rupa untuk memenuhi sedekat mungkin set tujuan yang diinginkan [12]. Definisi yang lebih umum adalah menugaskan satu set peristiwa (kuliah, kendaraan, acara-acara publik, dll) dengan set terbatas sumber dari waktu ke waktu sedemikian rupa untuk memenuhi kendala (batasan atau *constraint*) yang telah ditetapkan. kendala ini dapat dikategorikan sebagai *hard constraint* dan *soft constraint*, di mana *hard constraint* memiliki prioritas yang lebih tinggi dari pada *soft constraint*. Terdapat dua batasan dalam penyusunan penjadwalan kuliah, yaitu : *hard constraint* (harus terpenuhi atau batasan dari suatu permasalahan tidak boleh dilanggar) dan *soft constraint* (batasan dari suatu permasalahan diupayakan untuk terpenuhi) [22]. *Hard constraints* merupakan batas-batas yang harus diterapkan pada penjadwalan mata kuliah dan harus dipenuhi. Berbeda dengan *hard constraint*, *soft constraint* merupakan kendala yang tidak selalu dapat terpenuhi dalam proses pembentukan jadwal, akan tetapi meskipun tidak harus terpenuhi, jadwal yang dihasilkan harus semaksimal mungkin berusaha memenuhi ketentuan *soft constraint*.

2.2 Metode Heuristik

Metode heuristik merupakan suatu teknik untuk penyelesaian permasalahan yang tidak menekankan pada pembuktian apakah solusi yang didapatkan adalah benar, tetapi lebih menekankan pada performa komputasi dan kesederhanaan [18]. Metode heuristik merupakan suatu metode penyelesaian yang menggunakan konsep pendekatan. Pendekatan heuristik menggunakan suatu algoritma yang secara interaktif akan menghasilkan solusi yang akan mendekati optimal. Pendekatan heuristik menghasilkan perhitungan yang cepat karena dilakukan dengan membatasi pencarian dengan mengurangi jumlah alternatif yang ada. Pendekatan heuristik lebih dapat diterapkan pada permasalahan yang melibatkan jumlah input data yang besar dan kompleks.

2.3 Algoritma Genetika

Algoritma genetika merupakan teknik pencarian yang dilakukan atas sejumlah solusi yang mungkin yang dikenal dengan populasi [20]. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom berupa 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 kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga 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 [3]. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

Berikut merupakan contoh pseudocode algoritma genetika:

```

begin
    t <- 0;
    inisialisasi P(t);
    evaluasi P(t);
    while (not syarat berhenti) do
        begin
            kombinasi-kembali P(t) untuk membentuk C(t);
            evaluasi C(t);
            pilih P(t + 1) dari P(t) dan C(t);
            t <- t + 1;
        end
    End;
P : populasi
t : generasi / iterasi
C : offspring / anak

```

Gambar 2.1 Pseudocode Algoritma Genetika [20]

2.4 Algoritma Tabu Search

Tabu Search merupakan suatu metode optimasi yang menggunakan *short-term memory* untuk menjaga agar proses pencarian tidak terjebak pada nilai *optimum local*. Metode ini menggunakan *Tabu List* untuk menyimpan sekumpulan solusi yang baru saja

dievaluasi [21]. Selama proses optimasi, pada setiap iterasi, solusi yang akan di evaluasi akan dicocokkan terlebih dahulu dengan isi *Tabu List* untuk melihat apakah solusi tersebut sudah ada pada *Tabu List*. Apabila solusi tersebut sudah ada pada *Tabu List*, maka solusi tersebut tidak akan dievaluasi lagi pada iterasi berikutnya. Apabila sudah tidak ada lagi solusi yang tidak menjadi anggota *Tabu List*, maka nilai terbaik yang baru saja diperoleh merupakan solusi yang sebenarnya.

Secara umum berikut merupakan contoh pseudocode dari algoritma tabu search[14]:

```
1. s ← s0
2. sBest ← s
3. tabuList ← []
4. while (not stoppingCondition())
5. candidateList ← []
6. bestCandidate ← null
7. for (sCandidate in sNeighborhood)
8. if ( (not tabuList.contains (sCandidate))and(fitness(sCandidate)
> fitness(bestCandidate)) )
9. bestCandidate ← sCandidate
10. end
11. End
12. s ← bestCandidate
13. if (fitness(bestCandidate) > fitness(sBest))
14. sBest ← bestCandidate
15. end
16. tabuList.push(bestCandidate);
17. if (tabuList.size > maxTabuSize)
18. tabuList.removeFirst()
19. end
```

2.5 Algoritma Genetika - Tabu Search

Kombinasi dari algoritma genetika dan tabu search bertujuan untuk mengurangi kekurangan yang ada pada kedua algoritma. Kombinasi ini dimulai dari menciptakan solusi secara acak dengan beberapa iterasi. Algoritma tabu search digunakan untuk melengkapi algoritma genetika untuk mencapai suatu solusi, dan menciptakan suatu generasi solusi baru [10].

2.5.1 Keunggulan dan kelemahan algoritma genetika-tabu search

- Keunggulan:

- Algoritma genetika : Algoritma ini cocok untuk permasalahan yang kompleks yang biasanya diselesaikan dengan metode konvensional, dan tidak membutuhkan banyak perhitungan matematis terkait nilai fitness functionnya.
- Algoritma tabu search : Terdapat tabulist yang berfungsi untuk menyimpan kromosom yang sudah terpilih agar tidak terjadi redundansi data, dan berfungsi untuk mempertahankan nilai fitness agar tidak turun.
- Kelemahan:
 - Algoritma genetika : Dalam pengambilan data untuk pembentukan kromosom bersifat random sehingga solusi yang dihasilkan berbeda-beda setiap running.
 - Algoritma tabu search : Diperlukannya defisini alokasi memori sebelum algoritma ini dijalankan.

2.5.2 Perbandingan Penelitian Penulis dengan Peneliti Lainnya

Kombinasi algoritma genetika dan tabu search dalam pembuatan penjadwalan kuliah menggunakan uji coba sebanyak 30 mata kuliah dan 6 ruangan, sedangkan penelitian penulis ini menggunakan lebih dari 30 mata kuliah dan 6 ruangan sehingga permasalahan penjadwalannya semakin kompleks dan metode pada saat pengambilan data pun menggunakan metode yang berbeda [21]. Metode yang digunakan pada penelitiannya menggunakan metode *neighbourhood search* yang merupakan menemukan sebuah penaksiran untuk nilai minimum namun tidak memperhatikan tingkat keterjadwalan atau *constraint* yang dibentuk tidak begitu diperhatikan pada tahap inisialisasi. Sedangkan pada penelitian penulis ini sangat memperhatikan tingkat keterjadwalan sehingga akan memudahkan pada tahap cross over dan mutation dengan jumlah mata kuliah dan ruangan yang lebih kompleks. Terdapat beberapa keterbaruan dari penelitian ini, diantaranya:

- Pembuatan penjadwalan bisa dibuat dari awal penjadwalan maupun melanjutkan penjadwalan yang sudah ada.
- Mampu menjadwalkan dengan jumlah data yang besar.
- Mampu membuat jadwal pengganti apabila libur dalam 1 hari penuh.
- Dosen bisa memilih ruangan dan waktu tertentu.
- Waktu penjadwalan bisa diatur sesuai kebutuhan.
- Mampu menambahkan dosen, waktu, mata kuliah yang baru.