

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil yang didapat dalam pembuatan tugas akhir ini, maka dapat ditarik kesimpulan yaitu sebagai berikut :

1. Untuk melakukan penjadwalan kuliah dengan menggunakan kombinasi algoritma genetika dan *tabu search*, dapat dilakukan dengan cara menempatkan *tabu search* pada fase reproduksi yang bertujuan untuk menyeleksi kromosom agar tidak mengalami perkawinan silang secara berulang-ulang. Kromosom yang pernah mengalami perkawinan silang dalam fase reproduksi akan mengalami mutasi tanpa melalui proses perkawinan silang. Untuk *tabu tenure* pada *tabu search* dapat menggunakan metode pembebasan suatu *tabu list* dengan berdasarkan kondisi ukuran minimum memori komputer yang tersedia pada saat itu.
2. Kombinasi algoritma genetika dan *tabu search* menghasilkan jumlah iterasi yang lebih sedikit jika dibandingkan dengan tanpa menggunakan kombinasi *tabu search*. Namun dari sisi waktu proses, waktu yang dibutuhkan program untuk melakukan satu lompatan iterasi akan semakin lama. Hal ini terjadi dikarenakan ukuran *tabu list* dalam memori akan semakin besar sehingga proses pengecekan *tabu* suatu kromosom yang secara sekuensial akan membutuhkan waktu yang semakin lama pula. Oleh karena itu dari sisi waktu proses, penggunaan kombinasi ini akan menjadi kurang baik jika dibandingkan dengan tanpa kombinasi *tabu search*.
3. Penerapan algoritma *binary search* pada proses pengecekan kromosom dalam *tabu list* dapat meningkatkan waktu pencarian secara maksimal walaupun ukuran *tabu list* semakin besar. Sehingga dengan metode optimasi ini membuat kombinasi algoritma genetika dan *tabu search* menjadi lebih baik dari pada tanpa menggunakan kombinasi *tabu search*. Selain itu untuk proses pembuatan populasi awal pada

algoritma genetiknya, kromosom harus di-*filter*. Hal ini dilakukan dengan mencegah adanya kromosom yang sama didalam populasi awal tersebut sehingga variasi kromosom didalam populasi awal tersebut lebih bervariasi dan dapat membantu mencakup area solusi yang lebih luas. Sedangkan untuk optimasi dengan menggunakan konsep *multi threading* kurang cocok karena konsep kombinasi ini menggunakan satu *tabu list* sehingga akan membuat masing-masing objek *thread* menjadi saling berebutan untuk mengakses *tabu list* dalam satu memori sehingga tidak menghasilkan suatu optimasi.

6.2 Saran

Dari kesimpulan diatas ada beberapa saran yang dapat membantu mengatasi kekurangan dari perangkat lunak penjadwalan kuliah ini, yaitu meningkatkan optimasi penjadwalan kuliah dengan mengembangkan optimasi *multi threading* yang memiliki mekanisme yang dapat menghindari terjadinya perebutan akses kedalam *tabu list* oleh objek *thread*.