

## BAB 6 PENUTUP

Pada bab ini dibahas mengenai kesimpulan dari tugas akhir ini pada subbab 1.1, dan saran untuk pengembangan lebih lanjut mengenai penyelesaian penjadwalan kuliah pada subbab 1.2.

### 6.1 Kesimpulan

Dari penelitian tugas akhir ini dapat disimpulkan beberapa hal sebagai berikut.

1. GA dapat menyelesaikan masalah penjadwalan kuliah sesuai dengan ruang lingkup tugas akhir ini. Hal ini karena untuk *test case* yang dicobakan yaitu *test case* kecil, sedang, dan besar gasal, GA dapat menghasilkan jadwal yang memenuhi semua HC dan SC1 - SC5 (solusi optimal). Sementara, pada *test case* besar genap, GA tidak dapat menghasilkan jadwal yang memenuhi semua HC dan SC1 - SC5. Namun, jadwal yang dihasilkan sudah maksimal, yaitu dengan 24 pelanggaran pada HC2 karena pada *test case* besar genap memang tidak bisa ditemukan jadwal yang memenuhi semua HC dan SC1 - SC5 (lihat subbab 5.5).
2. Mengenai parameter GA untuk masalah penjadwalan kuliah dapat disimpulkan:
  - algoritma SPEA2 lebih baik dari *aggregation based* (lihat subbab 5.1.1),
  - representasi DRSSRSK dan DRSSRS lebih baik dari representasi lain (lihat subbab 5.1.2),
  - populasi semakin besar semakin baik (lihat subbab 5.3.1),
  - *archive size* yang ideal adalah 50% dari jumlah populasi (lihat subbab 5.3.2),
  - *crossover type one* dan *two* tidak jauh berbeda (lihat subbab 5.4),
  - *mutation rate* sangat tergantung dari panjang *genome* karena *mutation rate* yang baik untuk *test case* besar genap belum tentu baik untuk *test case* kecil (lihat subbab 5.4 dan 5.5).

## 6.2 Saran

Masih banyak hal yang dapat dilakukan untuk mengembangkan penelitian tugas akhir ini. Berikut saran-saran yang dapat dipertimbangkan untuk meneruskan penelitian ini.

1. Menggali lagi *requirement* kebutuhan penjadwalan yang ada di Fasilkom UI sehingga dapat dibuat jadwal yang diinginkan semua pihak (terutama dosen dan mahasiswa).
2. Menambah eksperimen dengan parameter GA yang lain seperti *chunk size* dan *tournament size*.
3. Mengubah operasi genetik jika diperlukan untuk mengefisienkan GA.
4. Memecah SC6 menjadi beberapa *soft constraint*.
5. Merancang representasi yang lebih sesuai jika ada tambahan *constraint* baru.
6. Membuat aplikasi dengan user *interface* (bukan *console*) yang menggunakan ECJ sebagai *solver* agar lebih mudah digunakan.
7. Menambahkan informasi baru ke mata kuliah yaitu informasi program studi dan menyelesaikan penjadwalan kuliah lintas program studi.

