

## BAB V

### KESIMPULAN DAN SARAN

Pada bab ini akan dipaparkan kesimpulan yang diperoleh dari penelitian yang dilakukan dan saran untuk penelitian mengenai pembuatan *package* *metaheuristicOpt*.

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian, pengujian dan analisis terhadap *package* yang dibuat, maka dapat ditarik kesimpulan sebagai berikut:

1. Pengembangan *package* *metaheuristicOpt* berhasil dilakukan dengan mengikuti tahapan metode pengembangan perangkat lunak sequensial linier yang terdiri dari analisis, perancangan, *coding* dan *testing*. Hasil *package* yang telah dibuat berhasil diunggah dan dipublikasikan di CRAN dengan alamat <https://cran.r-project.org/package=metaheuristicOpt>.
2. Perbandingan semua metode berhasil dilakukan dan terlihat bahwa metode *grey wolf optimizer* dan *particle swarm optimization* mendominasi dengan nilai *error* paling kecil. Selanjutnya kedua metode tersebut sangat cepat dalam mencapai nilai optimum yang diharapkan. Secara keseluruhan, semua metode yang diimplementasikan pada *package* *metaheuristicOpt* memiliki nilai *error* rata-rata yang kecil. Semua metode memiliki hasil yang sangat kompetitif sebagai metode untuk menyelesaikan permasalahan optimasi. Faktor kecepatan waktu eksekusi merupakan bagian dari analisis performa. Dari segi kecepatan, *genetic algorithm* memiliki waktu eksekusi paling cepat dan *grasshopper optimization algorithm* membutuhkan waktu yang sangat lama. Kemudian perbandingan *package* *metaheuristicOpt* dengan *package* lain yang ada di CRAN juga membuktikan bahwa *package* *metaheuristicOpt* terbaik dengan mengimplementasikan sebelas algoritma dibandingkan *package* lain yang hanya mengimplementasikan 1-5 algoritma saja.

## 5.2 Saran

Adapun saran penulis untuk kepentingan penelitian atau pengembangan lebih lanjut adalah sebagai berikut.

1. Menerapkan pendekatan *parallel computing* untuk sebagian atau semua algoritma yang diimplementasikan pada *package* metaheuristicOpt. Hal ini bertujuan untuk mempercepat waktu komputasi. Dari hasil penelitian ini terlihat bahwa terdapat beberapa metode yang membutuhkan waktu komputasi yang sangat lama.
2. Menambahkan algoritma lain untuk permasalahan optimasi. Algoritma yang ditambahkan dapat berupa algoritma terbaru ataupun algoritma lain yang populer yang belum ada pada *package* metaheuristicOpt, seperti algoritma *Differential Evolution* beserta variannya.
3. Menulis ulang kode *package* dengan menggunakan bahasa C atau C++ untuk menambahkan kecepatan waktu eksekusi setiap fungsi.
4. Membuat fungsi khusus untuk keperluan *plot* data yang menunjukkan *convergence rate* untuk setiap algoritma dalam bentuk grafik. Hal ini bertujuan untuk memudahkan pengguna dalam melihat atau menampilkan data secara representatif.