

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Kesimpulan

Setelah melakukan penelitian dalam melakukan studi komparatif menggunakan algoritma *population-based metaheuristic* pada DNA *sequence assembly*. Maka dapat diambil kesimpulan sebagai berikut dari penelitian yang telah dilakukan.

1. Algoritma *population-based metaheuristic* dapat digunakan untuk kasus DNA *sequence assembly* menggunakan bahasa pemrograman R. Dalam kasus DNA *sequence assembly*, algoritma *population-based metaheuristic* memaksimalkan total *overlap* dari permutasi urutan fragmen DNA. Perhitungan *overlap* antara sekuens DNA dilakukan oleh algoritma Knuth-Morris-Pratt. Permutasi dengan total *overlap* terbanyak lalu digunakan untuk menggabungkan DNA sesuai dengan *threshold*.
2. Dari eksperimen yang dilakukan pada data *benchmark*, didapatkan bahwa AVOA menunjukkan kinerja terbaik dengan menghasilkan jumlah *overlap* terbanyak, yaitu 49,952 pada dataset dengan panjang 750 dan cakupan 25. Selain itu, AVOA juga paling efisien dalam hal waktu komputasi dibandingkan dengan algoritma lain pada semua dataset yang diuji. Meskipun PSO memiliki hasil yang hampir sebanding dengan AVOA baik dalam *overlap* maupun efisiensi waktu, HBA unggul dalam menghasilkan jumlah *contig* paling sedikit, khususnya pada dataset dengan panjang 750 dan cakupan 15, dengan hanya 6 *contig*.
3. Efektivitas dari algoritma *population-based metaheuristic* dalam kasus DNA *sequence assembly* cukup baik karena tidak perlu melakukan percobaan di setiap permutasinya. Namun perlu dilakukan improvisasi pada model komputasi. Bahasa pemrograman R bukanlah bahasa pemrograman yang cepat, sehingga diperlukan efisiensi dalam penulisan kodenya. Pada penelitian ini, komputasi yang sangat berat ada pada algoritma KMP, sehingga dibutuhkan bantuan *package* Rcpp untuk menjalankan kode KMP menggunakan bahasa pemrograman C++.

## 5.2 Saran

Dalam pelaksanaan penelitian ini, penulis menyadari bahwa masih banyak kekurangan yang dilakukan dalam penelitian ini. Adapun saran yang dapat dilakukan pada penelitian selanjutnya adalah sebagai berikut.

1. Penelitian ini hanya berjalan pada sistem *stand alone*. Penggunaan komputasi paralel akan sangat membantu dalam mempercepat waktu komputasi.
2. Data yang digunakan dalam penelitian ini relatif sedikit untuk memvalidasi performa dari model komputasi. Data yang lebih banyak mungkin dapat berpengaruh dalam memvalidasi metode mana yang terbaik
3. Algoritma *population-based metaheuristic* lainnya dapat digunakan sebagai pengganti 4 algoritma yang digunakan dalam penelitian ini.
4. Bahasa pemrograman level rendah seperti C++ atau Java mungkin dapat digunakan agar waktu komputasi lebih cepat.