

BAB V

SIMPULAN DAN SARAN

5.1 SIMPULAN

Dengan diselesaikannya penelitian ini didapatkan beberapa kesimpulan dan keuntungan antara lain sebagai berikut:

1. Berdasarkan jumlah data sebanyak 100 data, hasil perbandingan efisiensi untuk waktu eksekusi paling cepat adalah algoritma *Selection Sort* dengan *running time* rata-rata 1,5 ms / 100 data, 1 ms / 80 data, 1 ms / 60 data dan waktu eksekusi paling lama adalah algoritma *Insertion Sort* dengan *running time* rata-rata 2,56 ms / 100 data, 1,6 ms / 80 data, 1,5 ms / 60 data, dan 1 ms / 40 data. Sedangkan untuk penggunaan memori yang terpakai dari 100 data tersebut, algoritma *Selection Sort* menghasilkan penggunaan memori yang terpakai lebih sedikit yaitu sebesar 18.662 KB dan algoritma *Insertion Sort* menghasilkan penggunaan memori yang terpakai lebih besar yaitu sebesar 25.604 KB. Dari hasil analisa perbandingan efisiensi algoritma *Selection Sort* dan *Insertion Sort* didapatkan bahwa algoritma *Selection Sort* adalah algoritma terbaik yang efektif dan efisien dalam menangani masalah pengurutan data
2. Berdasarkan hasil uji *Black Box testing*, dari beberapa uji komponen yang dilakukan diantaranya : *device* 1 dengan android versi 8.0, *device* 2 dengan android versi 9.0 dan *device* 3 dengan android versi 10. Semua komponen uji berhasil dilakukan, yang membedakan adalah tampilan aplikasi dari masing-masing *device* karena perbedaan ukuran layar.
3. Aplikasi ini menampilkan media informasi untuk menentukan jarak *Pet Shop* terdekat dan terjauh, tabel dan grafik memori yang berfungsi untuk menentukan memori yang terpakai serta menampilkan grafik waktu.

5.2 SARAN

Saran yang diberikan sesuai dengan adanya penelitian yang telah dilakukan adalah:

1. Algoritma *Selection Sort* diharapkan kedepannya dapat dikembangkan agar tidak membutuhkan penggunaan beberapa *method* tambahan.
2. Algoritma *Selection Sort* diharapkan dapat lebih stabil dalam menangani masalah pengurutan data.
3. Algoritma *Insertion Sort* diharapkan dapat lebih efektif dalam mencari posisi yang tepat untuk elemen larik agar tidak membutuhkan banyak operasi.
4. Algoritma *Insertion Sort* diharapkan kedepannya dapat dikembangkan agar dapat menangani masalah pengurutan dengan *list* yang terbalik tanpa harus memindai mengganti seluruh bagian.