BAB 6
SIMPULAN DAN SARAN

6.1 Simpulan

Dari penelitian yang berjudul "Sistem Pencarian Rute Menggunakan Algoritma *Branch and Bound* dan Dijkstra" ini telah berhasil dilakukan. Adapun kesimpulan yang dapat diperoleh adalah sebagai berikut:

1. Aplikasi dapat menghasilkan rute perjalanan yang mendekati optimal dari permasalahan TSP dari titik-titik tempat yang dipilih maupun rute perjalanan terpendek dari permasalahan pencarian rute perjalanan terpendek dan menampilkannya pada peta yang terdapat pada aplikasi.
2. Hasil perbandingan pengujian algoritma *Branch and Bound* dan *Brute Force* dalam penyelesaian permasalahan TSP menunjukkan *Brute Force* memiliki hasil optimal, karena hasil total jarak tempuh *Branch and Bound* memiliki rata-rata 8,711% lebih jauh dibandingkan perhitungan *Brute Force*, namun *Branch and Bound* mempunyai waktu pemrosesan rata-rata 173.914,451 kali lipat lebih cepat dibandingkan *Brute Force*.
3. Hasil perbandingan pengujian algoritma Dijkstra dan *Brute Force* dalam penyelesaian permasalahan pencarian rute terpendek menunjukkan keduanya memiliki hasil optimal karena total jarak tempuhnya sama, namun Dijkstra mempunyai waktu pemrosesan rata-rata 861.558,222 kali lipat lebih cepat dibandingkan *Brute Force*.
4. Aplikasi dapat menampilkan posisi pengguna berada dan menampilkan lokasi tukang tambal ban, bengkel, pom bensin atau petramini, minimarket dan ATM dan tipe data lainnya di sekitarnya yang sudah terdaftar ke dalam basis data oleh *administrator*.
5. Aplikasi dapat mendaftarkan lokasi tukang tambal ban, bengkel, pom bensin atau petramini, minimarket atau ATM baru atau tipe data lainnya dengan menggunakan GPS yang terdapat pada perangkat pengguna untuk mengambil letak posisi pengguna berada.

6.2 Saran

Untuk pengembangan lebih lanjut terkait penelitian ini, sejumlah saran diberikan, diantaranya adalah:

1. Metode pencarian rute TSP maupun rute terpendek dapat menggunakan metode lain apabila ada yang bekerja lebih baik dan efektif dalam menghasilkan rute perjalanan yang optimal.
2. Menggunakan *Google Maps API* yang berbayar, sehingga batasan *element* dan jumlah tempat yang dapat dimasukkan pun dapat bertambah.
3. Menambahkan *constraint* atau batasan urutan tempat yang harus dikunjungi terlebih dahulu, sehingga pengguna dapat memasukkan syarat pengunjungan tempat, misalnya seperti setelah mengunjungi tempat A, tempat yang dikunjungi haruslah tempat B.