

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan penelitian yang sudah dilakukan oleh peneliti pada pengembangan *Rest API* menggunakan *framework expressjs*, pengujian *whitebox testing*, *blackbox testing* dan metode pengembangan *scrum* dengan melakukan 2 kali *sprint*, memberikan kesimpulan sebagai berikut:

1. Pengembangan *Rest API* yang sudah dilakukan menggunakan *framework expressjs* dapat melakukan penambahan, menampilkan, mengedit dan menghapus data pada basis data. *Rest API* yang sudah dikembangkan dapat digunakan tim *frontend* dan *mobile* untuk pengelolaan data dan monitoring daerah rawan narkoba.
2. Pengembangan *Rest API* menggunakan metode pengembangan *scrum* yang dimulai dari menentukan *product backlog* hingga *sprint retrospective* berhasil dilakukan. *Sprint* dilakukan 2 kali dalam pekerjaan *product backlog* dengan melakukan pengujian *whitebox* dan *blackbox testing*. Hasil dari pengujian setiap *product backlog* berhasil sesuai dengan skenario dan memiliki tingkat resiko yang rendah.
3. Pengujian pada *Rest API* menggunakan *whitebox testing* dapat disimpulkan semua *Rest API* memiliki tipe prosedur sederhana dan tipe prosedur yang terstruktur dengan baik dan stabil serta memiliki tingkat resiko rendah. Pengujian pada *blackbox testing* dapat disimpulkan berfungsi sesuai dengan skenario dan hasil yang diharapkan. Performa yang dihasilkan pada *Rest API* batas area map memiliki kecepatan 231,4 milisekon dengan data 100 indikator.

5.2 Saran

Berdasarkan kesimpulan penelitian yang telah dilakukan pada pengembangan *Rest API* dengan *framework expressjs*, pengujian *whitebox* dan

blackbox testing, serta penerapan metode *scrum*, dapat memberikan saran untuk penelitian berikutnya sebagai berikut:

1. Disarankan untuk melakukan monitoring data pada basis data ketika data semakin bertambah. Semakin bertambahnya data akan berpengaruh pada kecepatan performa *Rest API*. Monitoring yang dilakukan akan menentukan arah untuk pengembangan berikutnya.
2. Disarankan untuk melakukan pengujian secara kompleks dari tipe perangkat lunak tampilan *website* dan tampilan *mobile* agar *Rest API* dapat berfungsi dengan baik.
3. Disarankan untuk melakukan pengujian kembali apabila terdapat perubahan pada struktur kode.