

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil Implementasi dan pengujian yang telah dilakukan dapat disimpulkan bahwa, optimasi *maintainability* menggunakan metode *clean code* dapat meningkatkan *maintainability* pada pengujian *human testing*, meskipun *automation testing* mengalami penurunan nilai *Maintainability Index* namun pada *automation testing* dapat meningkatkan nilai *score* dari penilaian *developer* dan menurunkan waktu *maintenance*. Serta developer lain memberikan penilaian sangat mudah terhadap pembacaan, pemodifikasi *source code*, berikut adalah table hasil pengujian *human testing* dan *automation testing* dapat dilihat pada

Tabel 5. 1 Hasil Pengujian

No	Jenis Pengujian	Hasil Pengujian
1	<i>AutomationTesting</i>	Hasil Dari <i>automation testing</i> atau hasil dari perhitungan nilai <i>maintainability index</i> didapatkan rata-rata 51.08 dimana
2	<i>Human Testing</i>	Pengujian Dari hasil <i>human testing</i> didapatkan <i>score</i> rata rata 4.4 dengan rata rata waktu 83detik

5.2 Saran

Dalam melakukan pengembangan perangkat lunak, salah satu bentuk pemanfaatan konsep *clean code* adalah dengan mengikuti *code convention* dari bahasa pemrograman yang digunakan. Pada dasarnya, *clean code* hanyalah petunjuk yang bersifat umum dalam menulis *source code* sehingga penggunaan *clean code* dapat disesuaikan dengan kebutuhan pengembang Adapun saran terhadap penelitian ini ke depannya dapat sebagai berikut:

1. Menambahkan beberapa faktor *clean code* terhadap penelitian kedepannya.
2. Mengimplementasikan *clean architecture*.
3. Melakukan *testing* dengan *phpunit* dan *unitesting*.