

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Hasil pengujian menunjukkan terjadinya peningkatan nilai kualitas *maintainability* setelah dilakukan *refactoring* sesuai kaidah-kaidah *clean code* dan *design pattern*. Hal ini menunjukkan bahwa penerapan konsep *clean code* dan *design pattern* dalam menulis kode sumber perangkat lunak dapat membuat perangkat lunak jadi lebih bisa dipelihara dengan baik.

#### **5.2 Saran**

Adapun saran kedepan untuk penelitian ini adalah:

1. Peningkatan *maintainability* juga dilakukan terhadap *frontend* dari perangkat lunak ini.
2. Pengukuran kompleksitas kode sumber menggunakan *cognitive complexity*. Pengukuran kompleksitas kode sumber menggunakan *cognitive complexity* memberikan hasil yang berbeda dengan *cyclomatic complexity*, karena pada *cognitive complexity* terdapat beberapa perbedaan dalam perhitungannya. Adapun yang membedakan *cognitive complexity* dengan *cyclomatic complexity* dalam aturannya adalah [14], [15]:
  - a. Mengabaikan struktur yang memungkinkan beberapa pernyataan dapat dibaca secara singkat menjadi satu
  - b. Penambahan 1 kompleksitas untuk setiap *break* dalam kode yang memiliki aliran yang linier.
  - c. Penambahan kompleksitas ketika struktur *flow-breaking* bersarang.