

## BAB V

### KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran pengembangan sistem ke depan.

#### 5.1. Kesimpulan

Berikut merupakan kesimpulan yang diperoleh dari rangkaian proses pengembangan dan pengujian sistem yang telah dilakukan:

1. Tools otomatisasi untuk pembuatan diagram *flowgraph* dan *cyclomatic complexity* berhasil dikembangkan dengan bahasa Golang. Tools ini bekerja dengan *parser* untuk mendapatkan struktur logika dari kode program, kemudian hasil *parser* tersebut dipindai dengan *Abstract Syntax Tree* untuk membangun *flowgraph*. Kemudian, sistem menghitung *cyclomatic complexity* dari *flowgraph* yang telah dibangun tersebut. Hasil analisis kemudian divisualisasikan melalui Fyne.io.
2. Hasil pengujian memperoleh bahwa 72 fungsi berhasil menjalankan seluruh *independent path*, sementara 37 fungsi mengalami kegagalan pada beberapa *independent path*. Dari segi kompleksitas, 106 fungsi memiliki *cyclomatic complexity* di bawah 5, yang mengindikasikan risiko rendah, sedangkan 3 fungsi lainnya memiliki nilai berkisar 5-10, yang mengindikasikan risiko rendah juga, namun memiliki prosedur yang relatif lebih kompleks.
3. Dari 37 fungsi yang mengalami kegagalan, sebanyak 17 fungsi mengembalikan pesan keberhasilan namun data tidak tersimpan di *database* sehingga direkomendasikan untuk menambahkan validasi pada proses penyimpanan data, 15 fungsi mengirimkan pesan *error* tanpa sumber kesalahan yang jelas sehingga perlu setiap lapisan mengembalikan pesan *error* yang informatif hingga ke main *root*, dan 5 fungsi mengalami kesalahan internal *query* sehingga perlu melakukan refaktor pada *query* agar sesuai dengan skema dan struktur *database* yang ada.

#### 5.2. Saran

Pengembangan selanjutnya disarankan untuk mengotomasi penentuan *independent path* dan penyusunan *test case* yang terintegrasi *unit testing*. Sehingga pengujian *white box* dengan teknik *basis path* dapat dilakukan secara lebih efisien dan mengurangi kesalahan manual.

*Halaman ini sengaja dikosongkan*