

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 KESIMPULAN

Setelah melakukan penelitian pengujian *white box* pada Sistem Informasi *Monitoring Skripsi Program Studi Informatika Universitas Pembangunan Nasional Veteran Jawa Timur*, didapatkan beberapa kesimpulan yang diantaranya sebagai berikut :

- a) Dari pengujian yang dilakukan dapat disimpulkan langkah – langkah dalam pengujian *white box* menggunakan *basic path* dengan teknik *flowgraph notation*, *cyclomatic complexity*, dan *deriving test case* yang dilakukan, yaitu :
  - i. Membuat *flowchart* dari *source code* program yang akan diuji.
  - ii. Membuat *flowgraph* dari *flowchart* yang telah dibuat.
  - iii. Menentukan nilai dari notasi *flowgraph* yang terdiri dari : *node*, *edge*, *predicate node* dan *region*.
  - iv. Melakukan perhitungan *cyclomatic complexity* dengan 3 cara, yaitu
    - a.  $V(G) = E - N + 2$
    - b.  $V(G) = PN + 1$
    - c.  $V(G) = \text{Jumlah Region}$
  - v. Menentukan jalur independen dari hasil perhitungan *cyclomatic complexity*.
  - vi. Membandingkan hasil perhitungan *cyclomatic complexity* dengan tabel resiko.

- vii. Membuat *deriving test case* dan melakukan pengujian terhadap fungsi yang berhubungan dengan fitur program berdasarkan tabel *deriving test case* yang dibuat.
- b) Dari pengujian yang dilakukan pada 6 *folder controller* dari program yang diuji dengan total fungsi dari keenam *folder* berjumlah 73 dari keseluruhan perhitungan untuk *cyclomatic complexity* menggunakan 3 cara didapatkan hasil yang sama sehingga bisa dikatakan jika *script code* program dari 6 *folder controller* relevan.
- c) Dari total fungsi berjumlah 73 menurut tabel hubungan *cyclomatic complexity* dengan resiko didapatkan 55 fungsi masuk dalam kategori tingkat resiko rendah dengan tipe prosedur sederhana, 14 fungsi masuk kategori tingkat resiko rendah dengan tipe prosedur yang terstruktur dengan baik dan stabil, 2 fungsi masuk kategori tingkat resiko menengah dengan tipe prosedur yang lebih kompleks, 2 fungsi masuk kategori tingkat resiko tinggi dengan tipe prosedur yang kompleks dan kritis. Dikarenakan tipe prosedur sederhana dan tipe prosedur yang terstruktur dengan baik dan stabil memiliki tingkat resiko yang sama rendah maka jika digabung diperoleh 69 fungsi memiliki tingkat resiko yang rendah, 2 fungsi dengan tingkat resiko menengah dan 2 fungsi dengan tingkat resiko tinggi. Maka diperoleh persentase sebesar 94,5 % dengan tingkat resiko terhadap cacat atau *error* yang rendah dengan tipe prosedur sederhana maupun yang terstruktur dengan baik dan stabil, 2,75 % tingkat resiko menengah terhadap cacat atau *error* dengan tipe prosedur yang lebih kompleks, dan 2,75 % tingkat resiko tinggi terhadap cacat atau *error* dengan tipe prosedur yang kompleks dan kritis pada seluruh fungsi yang akan diuji.

- d) Dengan diketahuinya tipe prosedur dari yang sederhana hingga tipe prosedur yang rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji beserta tingkat resiko dari yang rendah ke yang tinggi maka dapat diketahui atau dideteksi darimana kemungkinan besar terjadinya cacat atau *error* terhadap *source code* fungsi program yang terdapat pada *folder controller* dari aplikasi yang diuji.
- e) Dari keseluruhan fungsi berdasarkan *source code* program pada *folder controller* dari aplikasi yang diuji mewakili setiap fitur yang ada pada aplikasi dengan beberapa fungsi terdapat kondisi *private* dimana bisa berjalan fungsi tersebut jika fungsi yang lain men – *trigger* fungsi yang terdapat kondisi *private*.
- f) Untuk skenario pengujian dari jumlah jalur independen yang didapatkan dari perhitungan *cyclomatic complexity* jika ditotal berjumlah 221 uji coba dari yang seharusnya 285 menurut teori yang dimana setiap jalur independen harus bisa dieksekusi namun untuk kondisi pengujian yang dilakukan peneliti terdapat fungsi yang dimana kondisi *if* – nya untuk memproses data yang *inputan* berasal dari sistem dan tidak dapat dikendalikan secara manual oleh penguji, serta kondisi *if* yang digunakan untuk mengecek apakah fungsi berjalan dengan baik atau tidak sehingga penulis tidak bisa melakukan hal seperti membuat fungsi yang sudah benar harus dibuat salah untuk memenuhi eksekusi dari jalur independen tersebut. Lalu diperoleh persentase dari pengujian yang dapat dilakukan terhadap pengujian yang harusnya dicapai berdasarkan jumlah jalur independen sebesar 77,5 %.
- g) Dari total pengujian yang berjumlah 221 didapatkan hasil *valid* yang dimana kondisi pengujian yang diharapkan dan hasil dari sistem sama sehingga tidak

ditemukannya *error* dalam eksekusi *script code* program sesuai dengan fungsi menu atau fitur yang ada pada aplikasi.

## 5.2 SARAN

Berdasarkan hasil yang diperoleh pada penelitian ini, penulis memberikan saran sebagai berikut :

- a) Diperlukan untuk melakukan pengujian *white box* menggunakan teknik selain dari *basic path* untuk memperoleh hasil yang lebih baik terhadap pengujian terhadap aplikasi ini dari sisi *internal* program serta teknik pengujian *black box* lainnya selain dari pengujian *black box* yang telah dilakukan oleh peneliti sebelumnya sehingga diperoleh hasil yang lebih maksimal dan saling melengkapi terhadap pengujian aplikasi ini dari sisi *eksternal* program aplikasi ini.
- b) Pengujian *white box* menggunakan teknik *basic path* dengan metode *flowgraph notation* dan *cyclomatic complexity* cocok untuk menemukan tingkat resiko terhadap cacat atau *error* dan tipe prosedur dari *source code* program yang akan diuji serta relevansi dari *script code* program namun untuk metode *deriving test case* tidak dianjurkan dilakukan jika di dalam *script code* program terdapat kondisi *if* untuk proses data yang *inputan* berasal dari sistem dan tidak dapat dikendalikan manual oleh penguji serta kondisi *if* untuk mengecek kondisi dari fungsi berjalan dengan benar atau salah.