

BAB V

KESIMPULAN

Kesimpulan

Penelitian yang dilakukan dengan judul “Analisis Perebandingan *Test Case Prioritization* Menggunakan Algoritma *Greedy* Dan Algoritma *Hill Climbing* Pada Regression Testing” telah berhasil dilakukan menggunakan *website* demoblaze.com sebagai objek pengujiannya, penelitian ini dilakukan dengan melakukan pengujian *Regression Testing* pada *website* demoblaze.com sebagai proses pengumpulan data, serta *Test Case Prioritization* sebagai teknik pengujiannya berdasarkan perhitungan nilai efektivitas *Average Percentage Fault Detection* (APFD) dari penggunaan 2 algoritma, yaitu algoritma *Greedy* dan algoritma *Hill Climbing*.

Regression Testing dilakukan dengan diawali oleh proses menganalisis kebutuhan dari *software* dengan tujuan agar *software tester* dapat memahami kebutuhan suatu sistem, proyek, ataupun produk, tahapan selanjutnya yaitu mengidentifikasi masalah dengan cara mencari tahu *behavior* dari suatu *software* agar *software tester* dapat menemukan celah-celah yang berkemungkinan akan ditemukan *bug/defect*, selanjutnya pembuatan skenario pengujian berdasarkan *behavior* dari *software* yang akan dilakukan pengujian, setelah skenario pengujian atau *test case* telah dibuat, maka langkah selanjutnya adalah proses *Regression Testing* yang dilakukan berdasarkan langkah-langkah pada *test case*, dan yang terakhir proses evaluasi hasil *Regression Testing* berdasarkan analisis kebutuhan awal dengan cara membandingkan *expected result* dengan *actual result*, dan menyimpulkan ketidaksesuaian hasilnya sebagai *bug/defect*.

Implementasi *Test Case Prioritization* dengan mencampurkan *Severity Rating* kedalam penggunaan algoritma *Greedy* dan algoritma *Hill Climbing* cukup mempengaruhi proses prioritisasi, mencampurkan *Severity Rating* kedalam penggunaan algoritma *Greedy* dan algoritma *Hill Climbing* bertujuan agar proses *Test Case Prioritization* dapat dilakukan dengan memprioritaskan *test case* yang memiliki *fault* paling kritis terlebih dahulu agar perbaikan pada *software* tetap mengutamakan *fault* yang menjadi fokus utama dari kebutuhan suatu sistem,

proyek, ataupun produk, selain itu proses *Test Case Prioritization* juga dilakukan berdasarkan nilai efektivitas dari pengurutan prioritasnya, hal ini bertujuan agar hasil *Test Case Prioritization* juga dapat memberikan daftar prioritas perbaikan pada *software* yang paling efektif untuk dilakukan oleh pihak pengembang.

Proses perhitungan nilai efektivitas pada *Test Case Prioritization* dapat dilakukan menggunakan rumus perhitungan *Average Percentage Fault Detection* (APFD) yang proses menentukan kebutuhan data perhitungannya diambil berdasarkan fn yang pertama kali ditemukan pada urutan prioritas, setiap fn yang ditemukan pertama kali pada urutan prioritas 1 akan dimasukkan kedalam rumus dengan poin 1, sedangkan fn yang ditemukan pertama kali pada urutan prioritas 2 akan mendapatkan poin 2, begitupun dengan fn yang ditemukan pertama kali pada urutan prioritas 3 akan diberikan poin 3, dan seterusnya. Penggunaan perhitungan *Average Percentage Fault Detection* (APFD) sebagai penentuan nilai efektivitas juga dapat disimpulkan bahwa semakin banyak fn yang pertama kali ditemukan pada prioritas 1, maka nilai efektivitasnya akan semakin tinggi.

Penelitian ini menunjukkan hasil akhir bahwa penggunaan algoritma *Hill Climbing* pada proses *Test Case Prioritization* memiliki keunggulan secara nilai efektivitasnya dibandingkan dengan algoritma *Greedy*, algoritma *Hill Climbing* mendapatkan nilai efektivitas sebesar 0.91 sedangkan algoritma *Greedy* mendapatkan nilai efektivitas yang lebih rendah meskipun tidak secara signifikan, yaitu dengan nilai efektivitas sebesar 0.87.

Saran

Setelah melakukan pengujian serta menghadapi segala permasalahan selama penelitian ini berlangsung, saran penulis untuk penelitian selanjutnya yaitu:

1. Penggunaan algoritma untuk proses prioritisasi dapat dilakukan dengan menambahkan perbandingan, atau membandingkan algoritma lainnya seperti menggunakan algoritma *Genetic* maupun algoritma *Firefly*.
2. Penggunaan *software* sebagai objek pengujian *Regression Testing* juga dapat dilakukan pada aplikasi maupun *website* milik perusahaan yang aktif digunakan dalam kehidupan sehari-hari agar data *test case*-nya lebih banyak

dan lebih bervariasi sehingga terlihat dampak secara *real* pada penerapannya.

3. Perhitungan nilai efektivitas pada hasil prioritas dapat menggunakan perhitungan nilai efektivitas lainnya seperti APFDc, maupun APXC.