

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perangkat Lunak merupakan instruksi atau program komputer yang bila dieksekusi dapat menjalankan fungsi tertentu [1]. Untuk mendapatkan perangkat lunak dengan kualitas yang bagus maka dalam tahapan pengembangan sebuah perangkat lunak harus melalui proses penjaminan mutu. Salah satu aktifitas dalam penjaminan mutu perangkat lunak adalah pengujian perangkat lunak.

Pengujian perangkat lunak adalah sebuah proses, atau serangkaian proses yang dirancang untuk memastikan bahwa program telah berjalan sesuai dengan yang diinginkan [2]. Salah satu aspek penting dalam pengujian adalah pembangkitan kasus uji. Dalam pembangkitan kasus uji terdapat beberapa teknik yang digunakan diantaranya *Whitebox testing* dan *Blackbox testing*. *Whitebox testing* merupakan pengujian yang didasarkan pada kode program suatu perangkat lunak, namun pembangkitan kasus uji dengan cara ini relatif sulit diterapkan pada perangkat lunak yang kompleks. Sedangkan *Blackbox testing* merupakan pengujian yang tidak didasarkan pada kode program melainkan melainkan dapat membangkitkan kasus uji dengan menggunakan spesifikasi kebutuhan perangkat lunak.

Pada teknik *Blackbox testing* terdapat beberapa metode yang dapat digunakan untuk membangkitkan kasus uji diantaranya *Equivalence Partitioning*, *Boundary Value Analysis*, dan *Cause Effect Graphing*. Metode *Equivalence Partitioning* merupakan pengujian yang membagi data masukan dari perangkat lunak menjadi partisi data yang kemudian diturunkan menjadi kasus uji [3]. Metode *Boundary Value Analysis* merupakan pengujian yang berfokus pada batas dimana batas nilai – nilai ekstrim dipilih [3].

Dan metode *Cause Effect Graphing* adalah metode pengujian yang membantu dalam membangkitkan kasus uji berdasarkan pada hubungan antar *causes (input)* dan *effect (output)* yang terdapat pada spesifikasi kebutuhan perangkat lunak [4]. Namun, metode *Cause Effect Graphing* ini relatif lebih unggul dibandingkan metode lainnya dikarenakan metode ini memerhatikan integrasi antar kombinasi *input* dan *output* dan dapat mereduksi kasus uji. Oleh karena itu pada Tugas Akhir ini dibangun sebuah aplikasi pembangkit kasus uji yang mengimplementasikan metode *Cause Effect Graphing*.

1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, masalah-masalah yang dirumuskan antara lain:

1. Bagaimana mengimplementasikan metode *Cause Effect Graphing* untuk melakukan pengujian berdasarkan pada spesifikasi kebutuhan perangkat lunak?
2. Bagaimana kebenaran teknik *Cause Effect Graphing* dalam menghasilkan kasus uji?
3. Bagaimana teknik *Cause Effect Graphing* dalam mereduksi kasus uji?

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah:

1. Menerapkan metode *Cause Effect Graphing* untuk melakukan pengujian.
2. Menganalisis kasus uji yang dihasilkan dengan menggunakan metode *Cause Effect Graphing*.
3. Menganalisis kasus uji yang tereduksi dengan menggunakan metode *Cause Effect Graphing*.

1.4 Batasan Masalah

Adapun batasan masalah di Tugas Akhir ini sebagai berikut:

1. Pengujian dilakukan pada beberapa fitur aplikasi G - College yaitu Sign In, Sign Up level 1, Course, dan Update Status.
2. Pengujian dilakukan menggunakan tool yaitu Selenium Webdriver.
3. Spesifikasi kebutuhan perangkat lunak yang digunakan adalah yang memiliki deskripsi spesifikasi modul – modul aplikasi G-College.

1.5 Metodologi Penyelesaian Masalah

Penyelesaian Tugas Akhir ini dilakukan dengan metodologi sebagai berikut:

1. Studi Literatur
Pencarian referensi dan pendalaman materi terkait dengan pengujian *requirement*, metode *Cause Effect Graphing*, Selenium dan literatur lainnya yang berkenaan dalam penyusunan Tugas Akhir.
2. Pengumpulan Data
Data yang digunakan adalah deskripsi spesifikasi modul yang didapatkan dari Dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak). Spesifikasi tersebut akan digunakan untuk menghasilkan kasus uji yang digunakan untuk melakukan pengujian.
3. Pengembangan sistem
Mengembangkan sebuah sistem untuk menghasilkan kasus uji menggunakan metode *Cause Effect Graphing*. Dalam mengembangkan sebuah sistem dilakukan beberapa tahapan sebagai berikut:
 - a. Analisis
Melakukan proses pengumpulan informasi terkait kebutuhan sistem untuk mengetahui sifat program yang akan dibentuk.

- b. Perancangan
Melakukan perancangan sistem berdasarkan kebutuhan yang telah didefinisikan sebelumnya pada tahap analisis. Perancangan dilakukan dengan menggunakan pendekatan *Object Oriented*.
 - c. Pengujian
Melakukan pengujian terhadap sistem yang telah dibangun.
4. Analisis Luaran Eksekusi
Melakukan analisis pada kasus uji aplikasi G – College untuk mendapatkan hasil pengujian.
 5. Penyusunan Laporan
Melakukan penarikan kesimpulan dan melakukan penyusunan laporan Tugas Akhir yang diperoleh dari hasil penelitian dan dokumentasi.