



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam urutan proses pembangunan perangkat lunak, pengujian perangkat lunak adalah tahap yang dilakukan setelah implementasi atau pengkodean. Pengujian perangkat lunak atau *software testing* adalah proses melatih program dengan maksud mencari kesalahan awal sebelum program diberikan ke pengguna. Pada saat ini setiap setelah melakukan pembangunan atau pengembangan perangkat lunak akan selalu dilakukan pengujian perangkat lunak dengan teknik yang berbeda-beda. Contoh kasus yang terjadi, yaitu ketika pemerintah Inggris memperkenalkan pengajuan pengembalian pajak *online*. Pengguna terkadang dapat melihat berapa jumlah yang diperoleh oleh pengguna sebelumnya. Masalah ini terjadi karena perangkat lunak hanya diluncurkan tanpa dilakukan *software testing* (pengujian perangkat lunak) terlebih dahulu. *Developer* hanya fokus kepada perangkat lunak yang dibuat dan dapat membuat kesalahan pada perangkat lunak tersebut. Kesalahan yang mungkin terjadi karena adanya tekanan, kompleksitas sistem, dan perkembangan teknologi yang sangat cepat (Hambling, 2010).

Teknik perancangan *test case* untuk pengujian perangkat lunak dikelompokkan menjadi tiga kategori, yang pertama yaitu teknik pengujian perangkat lunak berdasarkan pada analisis tes dasar dokumentasi, termasuk aspek fungsional dan non-fungsional yang disebut *black box testing*. Yang kedua yaitu teknik pengujian perangkat lunak berdasarkan uji kasus langsung dari struktur komponen atau sistem, yang dikenal sebagai *structure-based* atau *white box testing*. Yang terakhir yaitu teknik pengujian perangkat lunak berdasarkan pada uji kasus dari pengalaman tester sistem serupa dan pengalaman umum dari pengujian, yang dikenal sebagai *experience-based techniques*. *Black box testing* memiliki kelebihan dibanding dengan kedua teknik di atas, yaitu dapat menentukan spesifikasi program di awal, tidak perlu melihat kode program secara detail, dan dapat digunakan untuk menilai konsistensi program. *Black box testing* adalah metode pengujian perangkat

lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal, dengan menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan (Hambling, 2010)..

Pada penelitian ini, akan dilakukan suatu pengujian menggunakan teknik *black box testing* yang difokuskan pada metode *Equivalence Partitioning* dan *Boundary Value Analysis* terhadap sebuah perangkat lunak. Pengujian akan dilakukan pada tingkat fungsional. Pengujian dilakukan dengan kedua metode tersebut karena keduanya relatif sederhana dibanding dengan tiga metode lainnya (*decision table testing*, *state transition testing*, dan *use case testing*) sebagai awalan. Kemudian setelah dilakukan pengujian akan dievaluasi berdasarkan kesalahan yang ditemukan atau *error* yang muncul dari sistem perangkat lunak tersebut.

Pada pengujian ini akan dilakukan sebuah pengembangan untuk perangkat lunak pertukaran kartu pos atau *postcrossing*. Dengan kembali maraknya kartu pos, banyak *user* yang memanfaatkan aplikasi *postcrossing* sebagai wadah pertukaran kartu pos dengan *user* lain di seluruh dunia. *Postcrossing* sendiri memiliki sebanyak 624,223 *member* dari 208 negara di dunia dengan *request* kirim 154 kartu pos tiap 1 jam (Postcrossing, 2004). Kekurangan aplikasi ini yaitu anggota terlalu lama dalam menunggu agar mendapatkan kuota mengirim kartu pos, dan *user* sering mendapatkan alamat negara yang sama karena banyaknya jumlah *user* dan aktifitas dari negara tersebut. Pada tugas akhir ini akan dilakukan *software development* dan *software testing* pada aplikasi *postcrossing* dengan pengujian menggunakan teknik *black box testing* yang difokuskan pada metode *equivalence partitioning* dan *boundary value analysis*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut maka dapat diambil tiga rumusan masalah, yaitu:

1. Bagaimana melakukan pengujian perangkat lunak dengan teknik *Equivalence Partitioning*?

2. Bagaimana melakukan pengujian perangkat lunak dengan teknik *Boundary Value Analysis*?
3. Bagaimana perbandingan dari kedua metode tersebut?

1.3 Batasan Masalah

1. Pengujian dilakukan pada 1 perangkat lunak, yaitu aplikasi *postcrossing*.
2. Teknik *black box testing* yang digunakan hanya ada 2, yaitu: *equivalence partitioning* dan *boundary value analysis*.
3. Perbandingan dilakukan dengan mencari perbedaan dari tahap pekerjaan dan karakteristik dari kedua teknik tersebut.

1.4 Tujuan Penelitian

Tujuan yang dapat dicapai adalah dapat mengetahui perbandingan antara kedua metode pengujian, yaitu *equivalence partitioning* dan *boundary value analysis* pada aplikasi *postcrossing*.

1.5 Manfaat Penelitian

Dari penelitian yang dilakukan mempunyai harapan agar aplikasi tersebut dapat bekerja dan beroperasi sesuai dengan keinginan dan kenyamanan *user* dan dapat bermanfaat sebagai dasar untuk penelitian pengujian berikutnya.

1.6 Sistematika Penulisan

Adapun sistematika penulisan laporan penelitian adalah sebagai berikut:

Bab I Pendahuluan

Pada bab ini akan dibuat pembahasan masalah penelitian secara umum berupa latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Kajian Pustaka

Bab ini berisi tentang penelitian yang sudah dilakukan sebelumnya dan digunakan sebagai dasar dalam proses analisis dan pengembangan perangkat lunak.

Bab III Metodologi Penelitian

Bab ini berisi rangkaian proses kegiatan yang dilakukan agar mendapatkan syarat-syarat untuk pengembang perangkat lunak.

Bab IV Pengembangan Perangkat Lunak

Bab ini berisi hasil analisis dari wawancara beberapa *postcrossers* untuk mendapatkan *software requirement* untuk pengembangan perangkat lunak. Kemudian akan dilakukan perancangan dan implementasi aplikasi.

Bab V Pengujian dan Pembahasan

Bab ini berisi tentang pengujian yang akan dilakukan dengan menggunakan 2 metode dari teknik *black box testing*, yaitu *equivalence partitioning* dan *boundary value analysis*. Setelah dilakukan pengujian, maka akan dibandingkan dengan *software requirement* dan dievaluasi.

Bab VI Kesimpulan dan Saran

Berisi tentang kesimpulan-kesimpulan yang merupakan rangkaian dari hasil penelitian yang telah diuraikan pada bab sebelumnya dan saran untuk pengembangan lebih lanjut pada penelitian tugas akhir ini.