

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di zaman sekarang berkembang sangat pesat. Perkembangan ini memberikan begitu banyak manfaat dalam berbagai aspek sosial. Teknologi dapat membantu manusia dalam mengerjakan pekerjaan-pekerjaan atau menyelesaikan masalah yang ada di sekitar manusia. Salah satu contoh teknologi dalam bentuk perangkat lunak yang dapat membantu masyarakat dalam mempermudah pekerjaan yaitu Osvas.

Aplikasi Osvas merupakan salah satu aplikasi yang dibuat untuk membantu manusia dalam pekerjaannya. Aplikasi Osvas merupakan aplikasi dikembangkan oleh PT Digdaya Olah Teknologi Indonesia. PT Digdaya Olah Teknologi Indonesia merupakan sebuah perusahaan *startup* yang bergerak dalam bidang teknologi informasi. PT Digdaya Olah Teknologi Indonesia berlokasi di kota Malang dan juga Jakarta.

Aplikasi Osvas sendiri terdiri dari dua aplikasi yang berbeda yaitu Osvas Properties dan Osvas Agent yang tentunya masing-masing aplikasi memiliki fungsinya tersendiri. Penelitian kali ini akan berfokus pada aplikasi Osvas Properties. Osvas Properties merupakan sebuah aplikasi jual, beli, dan sewa properti baik itu apartemen, kos, rumah, tanah dan masih banyak lagi. Osvas dibuat dengan tujuan untuk membantu serta memudahkan pengguna dalam membeli, menyewa serta menjual properti mereka. Tentu saja hal ini juga sangat membantu agen properti dalam menemukan klien pembeli properti. Aplikasi Osvas dibuat dalam *platform mobile* (android, iOS) dan juga tersedia dalam *website*, akan tetapi penelitian kali ini berfokus pada pengujian yang dilakukan pada aplikasi Osvas yang terdapat pada platform mobile lebih tepatnya operasi sistem iOS

Dalam proses pengembangan aplikasi Osvas yang dilakukan oleh tim *programmer*, tentu saja masih akan terdapat banyak *bug* atau *issue* yang ada

dalam setiap fungsi yang telah dibuat. *Bug* atau *issue* tersebut tidak akan langsung diketahui oleh tim *programmer*, maka dari itu dibutuhkanlah pengujian atau *testing* dalam proses pengembangan aplikasi Osvas. Aplikasi Osvas juga memiliki *bug* atau *issue* yang terdapat selama proses pembuatan. Hal ini tentu saja membuat aplikasi Osvas perlu dilakukan uji coba untuk memastikan bahwa aplikasi tersebut sudah terbebas dari *bug* atau *issue* yang dapat mempengaruhi jalannya sebuah fungsi dari aplikasi tersebut. Selain itu, tujuan dari dilakukannya proses *testing* ini adalah agar memastikan bahwa aplikasi Osvas yang nantinya akan di *deploy* sudah memiliki kualitas yang sesuai dengan standar aplikasi yang diinginkan oleh *customer*.

Proses *testing* aplikasi Osvas dilakukan oleh seorang *Quality Assurance (QA)*. *QA* sendiri memiliki tugas untuk memastikan bahwa aplikasi Osvas yang sedang dalam tahap pengerjaan tidak memiliki *bug* atau *issue* di dalamnya. Selain dari itu, tugas dari seorang *QA* adalah memberikan masukan kepada pihak *programmer* dalam hal *User Interface (UI)* jika dimana *UI* yang dibuat tidak sesuai dengan keinginan *customer*, singkatnya seorang *QA* dapat menjadi jembatan penghubung antara *customer* dan *programmer*. Dalam proses *testing*, dibutuhkan juga dokumen-dokumen penunjang yang perlu dipersiapkan seperti *test case*, *bug report*, dan *screenshot* ketika di temukannya *bug* atau *issue*, yang tentunya dokumen-dokumen tersebut dibuat oleh seorang *QA*.

Pengujian perangkat lunak dapat dilakukan dalam dua metode yang sudah tersedia. Metode – metode tersebut adalah manual testing dan juga *automation testing*. *Manual testing* merupakan metode pengujian yang dilakukan secara langsung atau bertatap muka dengan *software* yang dimana penguji atau *tester* berperan sebagai pengguna atau *user* pada umumnya, dimana *tester* melakukan pengujian terhadap semua fungsionalitas yang ada pada *software*. Dalam proses pengujian *manual* dibutuhkan tenaga yang cukup besar serta memerlukan ketelitian dan kesabaran dalam melakukan proses *testing*[1]. Berbeda dengan *manual testing* yang dilakukan tanpa sebuah *tools* atau *software*, *automation testing* merupakan proses pengujian

yang menggunakan *script testing* yang dimana *script testing* ini dijalankan dengan bantuan *software*[2].

Proses penelitian kali ini, penguji akan menggunakan metode *manual testing*. Alasan penulis menggunakan metode *manual testing*, yang pertama adalah karena *manual testing* dapat dijalankan tanpa adanya *script testing* dan yang kedua adalah karena *manual testing* lebih mudah dijalankan daripada *automation testing*. Selain itu, dalam sebuah penelitian tentang perbandingan *manual testing* dan *automation testing*, Dalam penelitian tersebut, di dapatkan bahwa selisih waktu yang diperlukan untuk *manual testing* adalah 61,47% yang dimana ini berarti bahwa pengujian *manual testing* lebih cepat 61,47 % jika dibandingkan dengan *automation testing*[3].

Black Box Testing merupakan salah satu metode pengujian baik itu dilakukan secara *manual testing* ataupun *automation testing*. *Black Box Testing* merupakan sebuah metode pengujian yang berfokus pada fungsionalitas *software* yang diuji. Alasan *Black Box Testing* sering digunakan dan populer dibandingkan metode *testing* lainnya adalah karena metode ini dapat beroperasi tanpa perlu bagi seorang *QA* untuk mengetahui isi dari sebuah code. Terdapat berbagai macam teknik dalam metode *Black Box Testing* yang dapat digunakan, seperti *Decision Table*, *All Pair testing* atau *Pairwise Testing*, *State Transition Table*, *Equivalence Partitioning*, dan *Boundary Value Analysis*. Dari 5 teknik yang terdapat dalam metode *Black Box Testing* yang telah disebutkan sebelumnya, *Equivalence Partitioning* merupakan salah satu teknik pengujian dalam metode *Black Box Testing* yang membagi data inputan dari sebuah *software* menjadi beberapa partisi dimana *test case* dapat diturunkan[4]. Dalam penelitian kali ini, penulis memilih menggunakan Metode *Equivalence Partitioning*, karena metode *Equivalence Partitioning* dianggap sebagai metode yang paling efisien dalam menemukan kesalahan fungsi[5]. Penelitian ini dilakukan hingga proses *developing* aplikasi *Osvas* selesai dan siap untuk di *deploy* ke *play store* atau *app store*.

1.2 Rumusan Masalah

Berdasarkan dari hasil latar belakang di atas, maka didapati rumusan masalah sebagai berikut:

1. Bagaimana hasil pengujian manual aplikasi Osvas Properties untuk transaksi properti menggunakan metode *black box testing*?
2. Berapa banyak waktu yang diperlukan untuk pengujian manual aplikasi Osvas Properties untuk transaksi properti menggunakan metode *black box testing*?

1.3 Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah yang diperoleh sebagai berikut:

1. Data yang akan digunakan pada penelitian kali ini merupakan data yang diperoleh dari proses *testing* aplikasi Osvas.
2. Teknik *black box testing* yang digunakan adalah *equivalence partitioning*
3. Yang dilaporkan dan diuji ini adalah pengembangan aplikasi Osvas Properties per tanggal 26 Agustus 2021 dengan versi #75
4. Penelitian ini dilakukan selama proses *developing* bukan saat proses *developing* selesai
5. Penelitian ini dilakukan untuk aplikasi Osvas Properties pada operasi sistem iOS
6. Penelitian dilakukan untuk *role tenant*

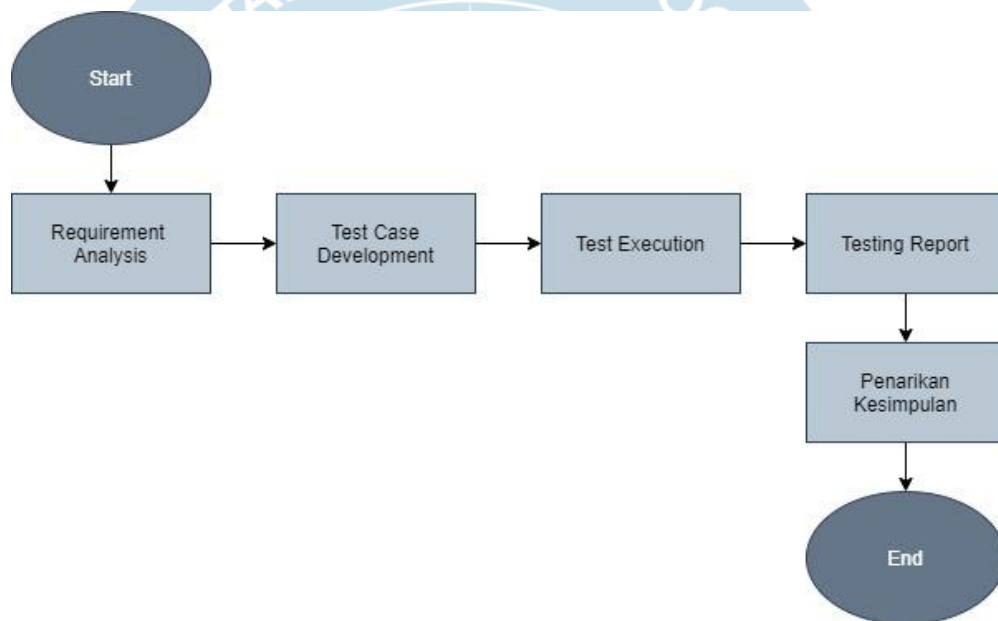
1.4 Tujuan Penelitian

Berdasarkan hasil dari latar belakang, maka penelitian ini memiliki tujuan sebagai berikut:

1. Mengetahui hasil pengujian manual aplikasi Osvas Properties untuk transaksi properti menggunakan metode *black box testing*.
2. Mengetahui waktu yang diperlukan untuk pengujian manual aplikasi Osvas Properties untuk transaksi properti menggunakan metode *black box testing*.

1.5 Metode Penelitian

Metode penelitian yang akan dilakukan pada penelitian ini adalah sebagai berikut:



Gambar 1.1 Metode Penelitian

1. *Requirement Analysis*

Requirement Analysis atau analisa kebutuhan adalah proses yang paling pertama dilakukan dalam penelitian kali ini. Pada tahap ini, penulis akan mempelajari alur dari sebuah fungsi yang kemudian akan diuji secara manual. Setelah analisa kebutuhan selesai, maka akan dilanjutkan ke tahap berikutnya yaitu *test case development*.

2. *Test Case Development*

Tahap *test case development* adalah tahap dimana akan dimulainya penyusunan dokumen *test case*. Pada tahap ini *test case* dibuat berdasarkan

setiap fungsi yang terdapat di aplikasi Osvas Properties. *test case* dibuat dalam bentuk tabel yang berisi komponen-komponen seperti id *test case*, skenario pengujian, variabel, *expected result*, keterangan, *output result*, keterangan, hasil pengujian. Ketika proses *Test Case Development* selesai, maka selanjutnya akan dilanjutkan ketahap *Test Execution*.

3. *Test Execution*

Test Execution adalah langkah selanjutnya dalam penelitian kali ini setelah proses *Test Case Development* dijalankan. Pada penelitian kali ini proses *Test Execution* dilakukan dalam bentuk *Manual Testing* dengan metode pengujian *Black Box Testing*. Pengujian untuk mencari *bug* dilakukan dengan cara yaitu dilakukan sesuai dengan *test case*. Ketika langkah *Test Execution* selesai, maka langkah atau metode penelitian berikutnya adalah *Testing Report*

4. *Testing Report*

Testing Report merupakan bagian akhir dari pengujian sebelum penarikan kesimpulan. *Testing Report* atau laporan pengujian berisi mengenai fungsi apa saja yang di uji dan fungsi apa saja yang berjalan dengan baik serta fungsi apa saja yang terdapat bug di dalamnya. Selain itu, dalam tahap *Testing Report*, penulis juga menuliskan presentase kesesuaian fungsi dalam bentuk persen (%). Waktu pengujian dari setiap fungsi juga dimasukkan dalam bentuk tabel, sehingga dapat memudahkan pembaca untuk mengetahui durasi pengujian dari setiap fungsi yang diuji.

5. **Penarikan Kesimpulan**

Pada tahap ini akan disajikan kesimpulan mengenai hasil dari penelitian ini. Hal ini diharapkan dapat memperoleh makna atau arti dari penelitian ini. Kesimpulan yang akan dihasilkan berupa bagaimana proses testing berjalan dengan menggunakan metode *black box testing* dengan *manual testing*. Selain itu, terdapat juga waktu pengujian selama pengujian berlangsung.

1.6 **Sistematika Penulisan**

Penulisan tugas akhir disusun dengan sistematika sebagai berikut

BAB I : Pendahuluan

Pada bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan tugas akhir.

BAB II : Tinjauan Pustaka

Bab ini berisi tentang penjelasan ringkas mengenai penelitian – penelitian terdahulu yang berhubungan atau memiliki kesamaan dengan penelitian yang dilakukan oleh penulis.

BAB III : Landasan Teori

Bab ini berisi mengenai penjelasan – penjelasan tentang teori yang menjadi landasan dari penelitian yang dilakukan oleh penulis. Teori-teori tersebut berupa *software testing life cycle*, *software testing*, *manual testing*, dan *black box testing*.

BAB IV : Analisis dan Perancangan Sistem

Bab IV berisi mengenai analisi kebutuhan dari penelitian dan juga berisi mengenai perancangan sistem. Dalam hal ini, bab 4 berisi tentang perancangan *test case* untuk proses pengujian yang dilakukan oleh penulis

BAB V : Hasil dan Pembahasan

Bab ini berisi dari hasil pengujian dan juga pembahasan mengenai hasil pengujian yang dilakukan oleh penulis. Hasil pengujian berupa *screenshot bug* yang ditemukan penulis selama penelitian berlangsung. Bab ini juga berisi mengenai dokumen *bug report* yang dibuat oleh penulis.

BAB VI : Kesimpulan dan Saran

Bab ini berisi mengenai kesimpulan dari penelitian yang sudah dilakukan oleh penulis. Bab ini juga menambahkan saran-saran yang berguna untuk penelitian yang lebih lanjut.