

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi informasi yang semakin pesat saat ini membawa perubahan signifikan dalam kehidupan manusia, terutama dalam hal penggunaan aplikasi web. *Website* semakin penting dalam kehidupan sehari-hari karena akses dan kegunaannya yang relatif mudah. Salah satu yang menentukan kualitas sebuah aplikasi *website* adalah kualitas dari *web user interface* (UI). Banyaknya *website* yang muncul membawa pengaruh yang besar bagi perkembangan informasi, namun pada waktu yang sama banyak insiden kegagalan terhadap teknologi tersebut sehingga menimbulkan kecemasan terhadap kualitas produk perangkat lunak tersebut [1].

Dalam konteks ini, penentuan dan pengimplementasian tuntutan kebutuhan yang jelas pada Spesifikasi Kebutuhan Perangkat Lunak (SKPL) sangatlah penting. SKPL atau biasa disebut *Software Requirement Specification* (SRS) akan menjadi dokumen acuan kerangka kerja yang menguraikan secara sistematis semua kebutuhan yang diharapkan dari perangkat lunak yang sedang dikembangkan [2]. Dengan demikian, SKPL akan memastikan bahwa fungsionalitas dari perangkat lunak akan memenuhi ekspektasi para pengguna dan pemangku kepentingan.

Antarmuka pengguna merupakan komponen perangkat lunak yang sering berinteraksi dengan *end-user*, oleh karena itu tahap evaluasi atau pengujian perlu dilakukan [3]. Pengujian dari komponen antarmuka *website* menjadi salah satu tugas dari proses penjaminan mutu perangkat lunak, yang mana proses pengujian sangat penting dan peran pengujian *software* tidak boleh dikesampingkan pada lingkup pengembangan perangkat lunak [4]. *(Graphical) User Interface test* berarti memvalidasi objek dari yang ditampilkan pada antarmuka, memeriksa alur fungsional dari objek antarmuka, serta memverifikasi data yang dihasilkan dari *backend* dan tampil dalam halaman *website* [5].

Menurut Anand Nayyar [6], banyak organisasi berpindah haluan menerapkan *automated testing*, karena hanya memanfaatkan seminimal mungkin waktu dan sumber daya. Alasan tersebut selaras dengan pernyataan Rafiq yakni pengujian harus memastikan kualitas produk, juga harus seefisien mungkin mulai dari waktu, sumber daya hingga biaya [7]. *Automated testing* yang mana merupakan praktik dari pengujian dengan alat kerja terotomasi sangatlah membantu pengembangan aplikasi untuk lebih efisien dari segi waktu dan sumber daya [1], [6]. Kasus uji akan lebih baik menggunakan bahasa alami, akan tetapi akan sulit jika akan diotomatiskan dengan mesin [5]. Pendekatan *keyword-driven testing* (KDT) berdasarkan *framework*-nya mendukung untuk pengujian otomatis dan memiliki manfaat lebih dalam hal pemeliharaan pengujian, kemudahan penggunaan, dan pemakaian ulang (tidak bergantung pada *tools*) [8], [9]. Representasi aksi yang dihasilkan oleh KDT menjadi dorongan tersendiri untuk menggunakannya dalam pengujian *website* ini.

Pengembangan kasus uji dengan KDT dibantu dengan pemanfaatan teknik seperti *equivalence class partitioning* (EQP), *decision table* (DT), dan *exploratory test* yang merupakan bagian dari metode *black box testing* untuk menghasilkan kasus uji. *Equivalence class partitioning* digunakan untuk menentukan kasus uji yang menjalankan fungsi atau perilaku yang sama pada kelasnya, dalam EQP hanya menggunakan satu kasus uji pada setiap partisi atau kelasnya sehingga lebih sedikit dari pada BVA [10]. Tabel keputusan digunakan pada fitur yang memerlukan kombinasi masukan yang dilakukan, pada penelitian milik Saeed, U. *et al.* [10] ditampilkan perbandingan antara BVA-EQP-DT yang mana tingkat kecanggihan DT dan EQP jauh lebih tinggi dibandingkan BVA. Metode *blackbox testing* berfokus pada perilaku eksternal sistem tanpa memperhatikan implementasi internalnya [10], [11]. Selain EQP dan DT, pembangkitan kasus uji secara eksplorasi digunakan untuk mendapatkan kasus uji yang memungkinkan untuk terjadinya *bug* atau kesalahan sistem.

Tel-U Kost atau Tekos merupakan sebuah startup bisnis rumah singgah atau indekos untuk mahasiswa dan orang sekitar Universitas Telkom. Dalam proses pengembangannya, tim pengembang Tekos memerlukan adanya pengujian yang menjamin fungsionalitas antarmuka sesuai dengan *requirement*-nya, salah satunya

yakni pengujian fungsional UI *website* guna menemukan *bug* atau *defect* sejak dini. Oleh karena itu pada kegiatan ini akan dilakukan uji fungsionalitas UI pada *website* Tekos dari sisi pemilik (FR1-n Tekos) dan pencari (FR2-n Tekos), menggunakan pendekatan KDT pada pengembangan kasus uji serta Katalon Studio dalam membantu pengujian terotomasi.

## 1.2. Perumusan Masalah

Dalam proses pengembangannya, tim pengembang Tekos memerlukan adanya pengujian yg menjamin fungsionalitas antarmuka. oleh karena itu, dirumuskan beberapa permasalahan yakni:

1. Bagaimana penerapan *keyword-driven testing* dalam pengujian fungsionalitas *user interface website* Tekos.
2. Apakah UI *website* Tekos sudah sesuai dengan *requirement*.

## 1.3. Tujuan

Penelitian ini memiliki tujuan utama yakni menganalisis kesesuaian antara antarmuka pengguna (UI) situs web dengan persyaratan yang ditetapkan untuk Tekos. Dalam upaya mencapai tujuan utama ini, *requirement* yang menjadi cakupan pengujian harus diuji dengan baik melalui pengembangan kasus uji dengan KDT yang tepat dan mengakomodir seluruh cakupan.

## 1.4. Batasan Masalah

Batasan kegiatan dari rumusan masalah tersebut yakni proses desain kasus uji didasarkan pada teknik-teknik *blackbox testing* seperti *equivalent class partitioning*, *decision table*, dan *exploratory testing*. Berikut merupakan fitur yang akan diuji.

- a. Sisi pemilik (owner) indekos, merupakan sub domain dari *website* Tekos yang berperan sebagai pemilik kos ataupun kontrakan. Pengujian dilakukan pada fitur-fitur berikut:
 

<ul style="list-style-type: none"> <li>- Daftar Akun (Account Registration)</li> <li>- Masuk Akun (Login Account)</li> </ul>	<ul style="list-style-type: none"> <li>- Lupa Password (Forget Password)</li> <li>- Keluar Akun (Logout Account)</li> <li>- Membuat Data Baru Kos (Input)</li> </ul>
--	--

- Perbarui Data Lama Kos (Edit)
  - Hapus Data Kos (Delete)
  - Membuat Data Baru Kamar (Input)
  - Perbarui Data Kamar (Edit)
  - Hapus Data Lama Kamar (Delete)
  - Membuat Data Baru Kontrakan (Input)
  - Perbarui Data Lama Kontrakan (Edit)
  - Hapus Data Kontrakan (Delete)
- b. Sisi pencari (*seeker*) indekos, merupakan domain utama dari *website* Tekos yang bertindak sebagai pencari. Pengujian dilakukan pada fitur-fitur berikut:
- Daftar Akun
  - Masuk Akun
  - Lupa Password
  - Keluar Akun
  - Pencarian Kos dan Kontrakan
  - Filter Pencarian Kos dan Kontrakan
  - Detail Produk Kos dan Kontrakan
  - Whatsapp atau kontak pemilik

### 1.5. Sistematika Penulisan

Guna memperjelas laporan ini, struktur penulisan laporan yakni sebagai berikut:

- a. BAB I PENDAHULUAN, menjelaskan mengenai motivasi dilakukannya kegiatan tugas akhir yang berkaitan dengan tema, menyangkut latar belakang, rumusan masalah, tujuan, dan batasan masalah dari tugas akhir.
- b. BAB II KAJIAN PUSTAKA, menjelaskan teori-teori dan fakta pendukung yang masih berkaitan dengan judul tugas akhir.
- c. BAB III METODOLOGI DAN ALUR PEMODELAN, berisikan metodologi yang digunakan serta alur kerja atau *framework flow*.
- d. BAB IV HASIL DAN PEMBAHASAN, bab ini menunjukkan hasil implementasi kegiatan tugas akhir dan analisa dari hasil implementasi tersebut.
- e. BAB V PENUTUP, berisikan kesimpulan dari kegiatan tugas akhir serta saran untuk peneliti selanjutnya berdasarkan tujuan, masalah, dan skop kegiatan.