

1. Pendahuluan

Latar Belakang

DKampus merupakan sebuah platform perangkat lunak berbasis *website* yang sedang dikembangkan agar dapat membantu para mahasiswa/i Telkom University dalam membeli makanan dan minuman tanpa perlu bepergian menuju lokasi. Tujuan dari platform ini adalah untuk memberikan akses kemudahan bagi para mahasiswa/i Telkom University dalam mencicipi jajanan yang ada disekitar kampus Telkom University tanpa harus datang ke lokasi karena platform ini menawarkan sistem *delivery online*. Selain mahasiswa/i, para UMKM sekitar kampus Telkom University dapat memasarkan dagangannya melalui platform ini dengan mudah.

Pada website DKampus ini belum dilakukan pengujian fungsionalitas, yang merupakan proses penting untuk memastikan bahwa setiap fitur dan fungsi pada website beroperasi sesuai dengan spesifikasi yang diharapkan. Tanpa pengujian fungsionalitas, ada risiko tinggi bahwa fungsi-fungsi inti dari *website* mungkin tidak berjalan dengan baik, yang dapat mengakibatkan *bug*, kegagalan sistem, atau ketidakmampuan *website* untuk memenuhi kebutuhan pengguna dengan benar. Penelitian sebelumnya menekankan pentingnya pengujian pada aplikasi *Action & Strategy* untuk memastikan keakuratan fungsional sistem dan pemenuhan kebutuhan pengguna [1], [2]. Selain itu, pesatnya perkembangan *website* dalam beberapa dekade terakhir telah mempengaruhi berbagai aspek kehidupan, dan penggunaan *website* sebagai platform pengembangan perangkat lunak menawarkan berbagai keuntungan [1], [3]. Oleh karena itu, kualitas perangkat lunak berbasis *website*, termasuk pengujian fungsionalitasnya, sangat penting untuk mendukung aplikasi tersebut.

Pengujian website mencakup pengujian fungsional, antarmuka, kegunaan, konfigurasi dan komabilitas, keamanan, serta performa. Pengujian ini berfokus pada aspek fungsional. Pengujian fungsional adalah pengujian yang dilakukan untuk memastikan bahwa fungsionalitas yang disediakan oleh aplikasi web sudah sesuai dengan kebutuhan [4]. Teknik pengujian fungsional dapat menggunakan analisis nilai batas, pengujian kelas ekivalensi, serta berbagai metode pengujian lainnya. Pengujian kelas ekivalensi atau yang biasa disebut *equivalence partitions* merupakan teknik pengujian perangkat lunak dengan membagi data masukan menjadi partisi-partisi yang *valid* dan *invalid*. Konsep pengujian menggunakan EP didasarkan pada persyaratan perangkat lunak. Metode ini membantu mengurangi waktu pengujian karena jumlah *test case* lebih sedikit [4]. Penelitian sebelumnya [5], [6] telah menunjukkan bahwa pengujian *black box* dengan teknik *equivalence partitions* efektif dalam mengidentifikasi kesalahan pada aplikasi perpustakaan, sehingga mengurangi risiko kerugian dan memastikan kualitas perangkat lunak dalam mengelola transaksi perpustakaan.

Terdapat dua metode dalam melakukan pengujian perangkat lunak antara lain pengujian manual dan otomatis. Dari kedua metode tersebut pengujian manual dapat memakan waktu relatif lama dan rentan terjadi *human error* sehingga pengujian otomatis dapat dijadikan solusi dalam menghemat waktu pengerjaan. Pengujian otomatis dapat secara otomatis menjalankan perintah yang telah ditetapkan [7]. Untuk melakukan pengujian otomatis diperlukan alat bantu/*tools*. Dalam hal ini, framework Katalon Studio dipilih sebagai solusi untuk pengujian otomatis. Pilihan ini didasarkan pada kemudahan penggunaan Katalon Studio yang dilengkapi dengan antarmuka *Integrated Development Environment (IDE)* yang intuitif. Katalon Studio memungkinkan pelaksanaan pengujian secara komprehensif terhadap berbagai jenis aplikasi, termasuk web, *mobile*, API dan desktop. Metode perekaman dan pemutaran ulang (*record and playback*) diterapkan untuk menghasilkan skenario pengujian yang efisien [8].

Pada pengujian perangkat lunak, terdapat dua teknik yaitu *white box testing* dan *black box testing*. *White box testing* merupakan pengujian yang berfokus pada pengecekan secara detail pada sumber kode yang dibuat.

Sedangkan *black box testing* adalah pengujian berdasarkan spesifikasi fungsional tanpa memeriksa desain dan kode program, bertujuan untuk memastikan bahwa fungsi, *input*, dan *output* perangkat lunak sesuai dengan spesifikasi yang ditentukan. Pengujian *black box* ini diuji dengan menerapkan teknik pengujian fungsional yang menguji setiap fitur dalam perangkat lunak untuk memastikan semua berfungsi sesuai dengan kebutuhan [9]. *Black box testing* memiliki keuntungan dibanding dengan *white box testing* antara lain penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, pengujian dilakukan dari perspektif pengguna, membantu mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.

Tujuan dari penelitian ini adalah memastikan fungsionalitas pada website DKampus dapat berjalan sesuai dengan *functional requirements*. Pengujian ini dilakukan menggunakan *black box* dengan teknik *equivalence partitioning* lalu dibantu oleh tools Katalon untuk melakukan pengujian secara otomatis. Pengujian ini berfokus pada fitur utama yaitu *login*, daftar alamat, *chat*, pengantaran, kelola UMKM, dan tambah produk/menu UMKM. Dengan pemilihan metode ini diharapkan dapat secara efektif menguji website DKampus dan website dapat berfungsi dengan baik.

Topik dan Batasannya

Berdasarkan latar belakang diatas, pengujian ini terbatas pada aspek fungsionalitas (*Functional Testing*) dari website DKampus, dengan fokus pada penggunaan teknik *Equivalence Partitioning* sebagai metode pengujian utama. Metode yang digunakan yaitu *Black Box Testing*. Pengujian dilakukan dengan bantuan alat otomatisasi Katalon Studio. Fokus utama pengujian adalah pada tiga fitur utama: *login*, daftar alamat, *chat*, pengantaran, kelola UMKM, dan tambah produk/menu UMKM.

Tujuan

Tujuan dari pengujian ini adalah untuk memastikan bahwa fitur-fitur utama dari website DKampus berfungsi sesuai dengan kebutuhan pengguna yang telah ditetapkan. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi keefektifan teknik *Equivalence Partitioning* dalam pengujian fungsional pada website DKampus. Hasil pengujian akan dianalisis untuk mengidentifikasi potensi perbaikan dan memastikan bahwa perangkat lunak memiliki kualitas yang baik.