

## BAB II TINJAUAN PUSTAKA

### 2.1. Studi Sebelumnya

Pada penelitian ini, peneliti mengambil konsep dari SDLC, SDLC adalah konsep manajemen proyek yang mengambil bentuk tahap pengembangan produk perangkat lunak dan bertujuan untuk memastikan bahwa produk mematuhi atau sesuai dengan rencana yang diberikan[8]. Pengujian adalah salah satu proses dalam SDLC. Pengujian mempunyai tahapan seperti validasi dan evaluasi perangkat lunak yang dilakukan dalam pengembangan. Selama pengujian perangkat lunak akan menghasilkan dua hasil: hasil yang diharapkan dan yang diinginkan. Penulis menjelaskan penelitian sebelumnya sebagai pembandingan dengan penelitian yang dilakukan oleh penulis.

Penelitian menggunakan metode pengujian *black box testing* yang diimplementasikan pada *website*. Metode ini berfungsi menilai kualitas fitur pada *software* dalam memenuhi kebutuhan pengguna. Kajian yang dilakukan dalam menilai kualitas perangkat lunak yang disebarkan berdasarkan aspek antarmuka pengguna. Pada pengujian yang telah dilakukan, pada *developer* perangkat lunak bisa meyakinkan para pengguna bahwa sistem dapat memenuhi harapan mereka. Hasil yang diperoleh dengan menerapkan konsep *cross browser testing*, mendapatkan hasil yang baik menurut pengalaman pengguna[9].

Penelitian pengujian dilakukan dengan metode *black box* pada aplikasi berbasis android, di mana penelitian menggunakan teknologi yang disebut *Phonegap*, yaitu aplikasi platform untuk menguji dan mengetahui fungsionalitas pada kebutuhan pengguna, dan berfungsi dengan baik dan sesuai dengan dokumentasi yang telah ada. Pada pengujian yang dilakukan, didapatkan bahwa bebas dari kesalahan pada *interface* dan transaksi[10]. Penelitian lain menerapkan metode pengujian yang berbeda, yaitu metode pengujian *white box* untuk memeriksa aliran pengodan *software*. Pengujian pada aplikasi menggunakan teknik *path-based* dalam mendapatkan hasil logika dalam program. Hasil pengujian pada fungsi yang bekerja dengan baik, tetapi memiliki logika dan alur program yang kurang optimal[11].

Pada pengujian yang lain, salah satu studi melihat tes yang dilakukan dengan perekaman dan pemutaran. Tujuan dari penelitian adalah untuk menganalisis kinerja perangkat lunak yang dibangun melalui pengujian otomatis menggunakan Katalon Studio. Berdasarkan penelitian ini, dapat disimpulkan bahwa secara umum pengujian perangkat lunak dapat dilakukan secara otomatis menggunakan *tools automation*. Katalon Studio telah bekerja sebagai alat otomatisasi yang berfungsi dengan baik untuk melaksanakan pengujian otomatis, meskipun masih terdapat beberapa kekurangan[12]. Setelah penelitian lain yang menguji aplikasi platform seluler menggunakan Katalon Studio, para peneliti juga menunjukkan hasil yang positif. Katalon Studio terintegrasi dengan baik dengan Microsoft Excel untuk melakukan pengurutan data yang digunakan selama pengujian. Selanjutnya, tes otomatis dapat dilakukan lebih efektif dari pada tes dengan manual[13].

Pengujian perangkat lunak dengan pengujian *black box* memiliki teknik yang dapat diterapkan dalam proses penelitian. Terdapat studi pengujian perangkat lunak pada aplikasi manajemen parkir di mana pengujian ini menggunakan pengujian *black box* dengan teknik ekuivalen *zoning*. Teknik ini membagi input ke dalam kelas data tertentu yang dapat mengidentifikasi input sebagai valid atau tidak valid. Para peneliti menyarankan persiapan studi yang cermat dan penggunaan teknik pengujian *black box* lainnya untuk menemukan lebih banyak variasi kesalahan, seperti nilai batas atau kesalahan tebakan, sehingga hasilnya dapat memenuhi harapan. Dengan ditemukannya varian *error* tersebut diharapkan kualitas aplikasi dapat ditingkatkan[14]. Berbeda dengan penelitian selanjutnya, tes sistem penilaian siswa menggunakan teknik analisis nilai batas, menerapkan uji batas atas dan batas bawah dari data input. Hasil yang diperoleh berdasarkan input dan *output* dari aplikasi menggunakan teknik analisis nilai batas yang baik. Namun peneliti menyarankan untuk menggunakan cara dan metode pengujian lain dalam mencapai hasil yang lebih optimal[15].

Berdasarkan uraian penelitian sebelumnya maka penulis akan melakukan penelitian terkait dengan sistem *website Job Fair* berbasis *website*. Penulis menggunakan pengujian secara manual dengan merancang *test case* dan pada pengujian secara otomatis menggunakan *tools automation* Katalon Studio. Tujuan dari pengujian ini untuk mengetahui apakah fungsi berjalan dengan baik dan sama seperti fungsionalitas yang ada. Selanjutnya digabungkan dengan *Partition Equivalence* (EP) yaitu pengujian berdasarkan nilai input yang valid dan tidak valid. Penulis merangkum perbandingan penelitian dalam tabel 2.1.

**Tabel 2.1 Perbandingan Penelitian Terdahulu**

No	Penelitian	Judul Penelitian	Objek Penelitian	Tujuan Penelitian	Jenis Pengujian	Hasil Penelitian
1	<b>Herlinda et al.</b>	<i>AUTOMATION TESTING TOOL</i> DALAM PENGUJIAN APLIKASI BELAJAR TAJWID PADA <i>PLATFORM ANDROID</i>	Aplikasi Belajar Tajwid berbasis Android	Menganalisis efektivitas aplikasi yang sudah dibangun, yang kemudian apakah <i>Automation Testing Tool</i> Katalon berdaya guna dibandingkan pengujian manual.	Pengujian otomatis	Aplikasi Katalon Studio masih memiliki kekurangan karena masih harus dikontrol oleh penguji ketika menguji perangkat <i>mobile</i> . Namun, sebagai alat pengujian otomatisasi, Katalon telah memenuhi kriteria fungsi yang diperlukan untuk pengujian otomatis.
2	<b>Fahmi Ahmad Fauzi et al.</b>	Pengujian Terhadap Aplikasi <i>Parking Management</i> Menggunakan Metode <i>Black-Box</i> Berbasis <i>Equivalence Partitions</i>	Aplikasi <i>Parking Management</i>	Menemukan kesalahan dan memperbaikinya sehingga sistem dikatakan layak untuk digunakan dan menyesuaikan fungsi-fungsi dengan spesifikasi yang diinginkan.	Pengujian manual	Berdasarkan pengujian kualitas aplikasi <i>parking management</i> , dapat ditarik kesimpulan bahwa pengujian dengan metode <i>Black box</i> berbasis <i>Equivalence Partitions</i> dapat membantu proses pembuatan <i>case</i> pengujian, uji kualitas dan menemukan kesalahan yang tidak disengaja.
3	<b>Wahyu Nur Cholifah et al.</b>	PENGUJIAN <i>BLACK BOX TESTING</i> PADA APLIKASI <i>ACTION &amp; STRATEGY</i> BERBASIS <i>ANDROID</i> DENGAN TEKNOLOGI <i>PHONEGAP</i>	Aplikasi <i>Action &amp; Strategy</i> berbasis Android dengan teknologi <i>Phonegap</i>	Mengetahui apakah fungsi-fungsi pada aplikasi telah berjalan sesuai dengan fungsinya dan mengevaluasi kesesuaian aplikasi dengan kebutuhan pengguna.	Pengujian manual	Dalam hasil akhir menggunakan <i>black box</i> , tidak ada bug atau kesalahan yang ditemukan pada proses pengujian fungsional untuk aplikasi atau situs web.

No	Penelitian	Judul Penelitian	Objek Penelitian	Tujuan Penelitian	Jenis Pengujian	Hasil Penelitian
4	<b>Meiselino Ansfridus</b>	Implementasi <i>Black box testing</i> pada Sistem <i>Techno Expertise Academy</i> (TEA) (Studi Kasus: Astra Credit Companies)	Aplikasi <i>web</i> (CMS) QMTEA Sistem <i>Techno Expertise Academy</i> (TEA) Astra Credit Companies	Mengetahui fungsi sistem TEA dapat bekerja dengan baik, dan menemukan kesalahan pada sistem TEA yang perlu segera diperbaiki, serta pengujian manual dan pengujian otomatis dapat mencapai hasil yang optimal.	Pengujian manual dan otomatis	Hasil penelitian dengan penerapan metode pengujian <i>black box</i> dengan teknik Partitioning Equivalence menghasilkan ketersediaan aplikasi 100%. Berdasarkan hasil yang diperoleh, fungsionalitas sistem TEA bekerja dengan baik tanpa kesalahan dan pengujian manual dan pengujian otomatis berhasil mencapai hasil yang optimal.
5	<b>Adi Purnomo</b>	<i>SOFTWARE TESTING</i> APLIKASI WEBSITE PT GRAMEDIA MENGUNAKAN METODE <i>BLACK BOX</i> PADA PT WGS BANDUNG	Aplikasi berbasis <i>website</i> PT.Gramedia Pustaka Utama	Menilai Kualitas pada pada <i>software</i> dengan metode pengujian <i>black box</i> pada aplikasi berbasis <i>website</i> pada PT Gramedia Pustaka Utama.	Pengujian manual	Hasil penelitian berdasarkan aspek <i>Functional Testing</i> yang meliputi input, proses, dan <i>output</i> pada situs <i>website</i> Gramedia adalah telah terpenuhi/sukses secara keseluruhan sesuai dengan skenario/ <i>user story</i> .

## 2.2. Dasar Teori

### 2.1.1 Pengujian Perangkat Lunak

Pengujian Perangkat lunak atau *software testing* adalah pengujian dalam mengetahui mutu dari *software* atau perangkat lunak dalam memenuhi persyaratan awal yang telah ditentukan atau tidak, Tujuan dari pengujian perangkat lunak adalah mencari kesalahan dalam sistem semaksimal mungkin, sehingga standar kualitas pada produk dapat terpenuhi. Biasanya pengujian dilakukan dengan mencari kesalahan pada sistem, kemudian mencatat hasil seperti kekurangan, kesalahan, atau mengidentifikasi cacat pada sistem, dan kemudian mengevaluasi hasil dari setiap fungsi yang akan dikembangkan. Mencari kesalahan pada sistem sangat penting, karena pada tidak ada sistem yang sempurna. Pengujian dilakukan dengan harapan dapat mengetahui apakah perangkat lunak telah memenuhi harapan dari penggunaan dari pengguna[16] .

### 2.1.2 Metode Pengujian Perangkat Lunak

#### 2.2.2.1. *Black box Testing / Functional Testing*



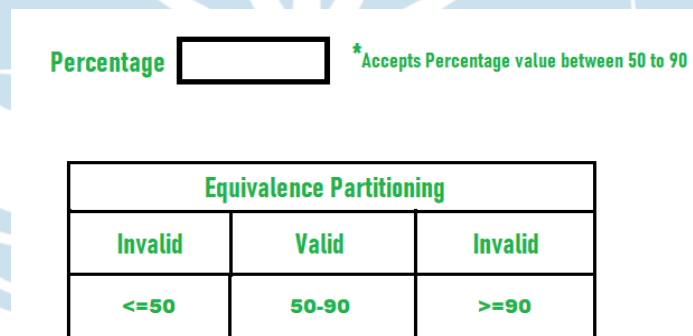
**Gambar 2. 1. Representasi *Black box testing***

*Black box testing* adalah pengujian yang berorientasi pada fungsional dalam sistem, yang artinya metodologi yang dibuat untuk dapat mengamati hasil dari nilai masukan tanpa melakukan pengodian sama sekali. Penguji tidak boleh memiliki kode sumber pada sistem itu sendiri. *Black box testing* hanya berfokus pada keluaran data yang dihasilkan sebagai respons terhadap masukan data dengan kondisi

yang telah ditentukan sudah tepat. Dalam pengujian perangkat lunak hanya mengetahui informasi yang dimasukkan ke dalam *Black box testing*, dan *black box testing* akan mengirimkan keluaran dari masukkan yang telah dimasukkan tersebut. Metode *black box* bertujuan untuk memenuhi harapan dari pengguna berdasarkan kebutuhan fungsional pada perangkat lunak. *Black box* juga dapat disebut *functional testing* karena hanya berfokus pada kebutuhan fungsional[17]. *Black box testing* biasanya melihat dari sudut pandang para pengguna di mana *user* langsung mencoba menggunakan *website* tersebut dan melihat apakah fungsi yang telah dicoba berjalan dengan baik atau tidak.

### 2.3. Teknik Pengujian *Black box testing*

#### 2.3.1. *Partitioning Equivalence*



**Gambar 2. 2. Representasi *Partitioning Equivalence***

*Partitioning Equivalence* adalah teknik pengujian nilai input dan *output* yang dihasilkan dalam suatu fungsi. *Partitioning Equivalence* berasal dari gagasan bahwa input ke program dapat dipartisi ke dalam kelas-kelas yang setara. Menguji seluruh kelas pada perangkat lunak dengan nilai apa pun yang di inputkan di dalamnya. Dalam implementasinya, beberapa peneliti menjelaskan bahwa dengan menggunakan nilai input apa pun, mereka dapat menemukan kesalahan fungsional yang ada di domain. Nilai input menghasilkan *output* dengan valid atau tidak valid sesuai keinginan



fungsional yang diinginkan oleh pengguna. Gambar 2.2 menunjukkan cara kerja kesetaraan partisi.

Teknik *Partitioning Equivalence* diimplementasikan dengan memartisi kelas dari nilai input ke dalam desain *Test Case*. Hasil yang valid atau tidak valid akan diperoleh berdasarkan batasan input yang muncul pada fungsi. Pengujian yang dilakukan cukup untuk membuat satu kondisi dari setiap kategori[18]. Keuntungan dari *Partitioning Equivalence* adalah memberikan rasa pengujian yang lengkap, termasuk nilai input dan *output* secara keseluruhan dengan kategori yang lebih kecil, dan menghindari *redundancy* atau pengulangan pada pengujian[19].

### **2.3.2. Manual Testing**

*Manual testing* atau pengujian manual adalah pengujian perangkat lunak yang dilakukan tanpa menggunakan alat. Pengertian lain pada pengujian manual adalah proses dalam memverifikasi desain dari sistem dan mengidentifikasi cacat berdasarkan kasus uji [20]. Dengan kata lain, pengujian manual dilakukan berdasarkan kasus uji yang dibuat dalam *test case* dan kemudian membandingkan nilai keluaran aktual dengan nilai keluaran yang diharapkan sesuai dengan harapan pengguna. Tes manual memiliki waktu pemrosesan yang lama dan lebih mahal dari pada metode pengujian lainnya. Keterbatasan waktu dan SDM dapat mempengaruhi berjalannya proses pengecekan hasil. Tetapi, dalam mendeteksi kesalahan pada perangkat lunak, pengujian manual sangat efektif untuk dilakukan. [21].

### **2.3.3. Automation Testing**

*Automation Testing* atau pengujian otomatis memainkan peran penting dalam sistem SDLC karena menguji dengan cepat, di mana pengujian otomatis menggunakan *automation tools* dalam pengujiannya. Cara ini merupakan langkah yang efektif dalam mengurangi usaha yang dilakukan secara berulang-ulang. Mengurangi upaya dan mempercepat siklus pengujian dalam melakukan pengujian secara manual[22]. Ini menjelaskan

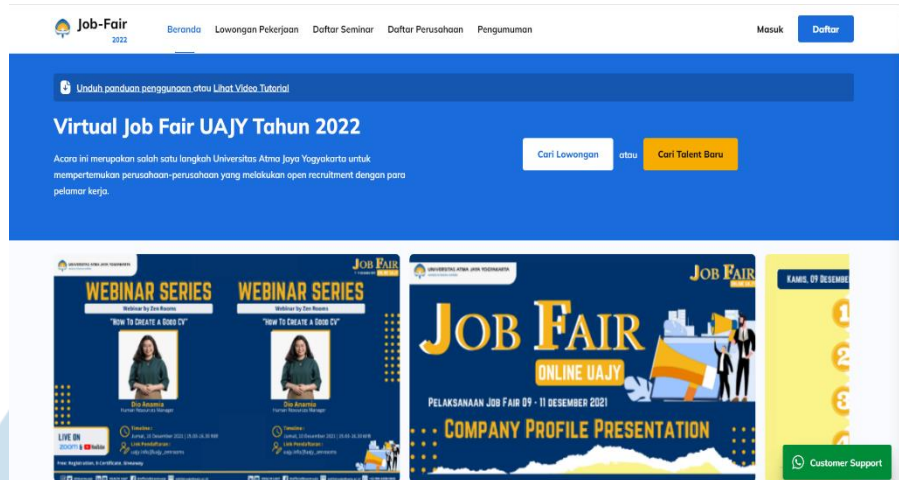
bahwa *Automation testing* dapat disebut pengujian yang dinamis. Efektivitas dalam penerapan alat otomatis tergantung pada identifikasi alat dan pemilihan kasus uji. Ketika proses *testing* berjalan dan melaporkan kesalahan, jika tidak melaporkan kesalahan, *testing* akan selesai dan *backlog* pengujian berakhir.

*Automation testing* digunakan untuk menguji secara fungsionalitas. Pada *automation testing* mempunyai kelebihan di mana terdapat *regression*, penghematan periode uji, eksekusi yang cepat, dan dapat dilakukan oleh semua orang. Sedangkan pada saat mengimplementasikan *automation testing* perlu adanya *support & maintenance* agar kualitas pengujian dapat terus terjaga. *automation tools* yang sering digunakan adalah Watir, *Unified Functional Testing*, Katalon Studio, *Testcomplete*, dan Selenium [23].



## 2.4. Website

### 2.4.1. Website Job Fair UAJY

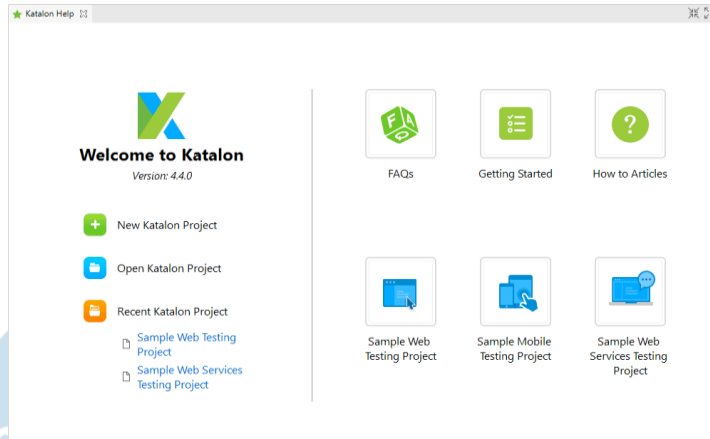


Gambar 2. 3. Website Job Fair UAJY

Website Job Fair UAJY adalah sebuah perangkat lunak berbasis website yang adalah salah satu sistem yang menunjang kinerja pada internal organisasi. Gambar 2.3 merupakan halaman awal yang akan muncul saat mengakses link website Job Fair UAJY. Website ini digunakan oleh semua pencari kerja di seluruh Indonesia dalam melamar pekerjaan. Website Job Fair bertujuan untuk membantu para pencari kerja dan perusahaan untuk dapat melakukan proses rekrutmen secara online dengan cara membuat aplikasi berbasis web sebagai wadah dalam mempertemukan para HRD dan pencari kerja. Pembagian akses pada sistem website Job Fair secara keseluruhan dibagi menjadi 3 bagian, pertama adalah sebagai pencari kerja, di mana akses ini diberikan kepada setiap pelamar yang ingin mengirimkan lamarannya di perusahaan yang terdaftar pada website Job Fair UAJY. Kedua adalah company, di mana akses ini berikan kepada setiap perwakilan perusahaan dalam menerima lamaran pencari kerja, ketiga adalah crew yang berfungsi mengelola semua data master pada user website Job Fair UAJY.

## 2.5. Tools

### 2.5.1. Katalon Studio



**Gambar 2.4 Tampilan Katalon Studio**

Gambar 2.4 adalah tampilan aplikasi Katalon Studio, salah satu *automation tools* untuk pengujian perangkat lunak. Katalon Studio dibuat oleh Katalon Inc pada tahun 2015, dibuat dengan *framework open-source* Selenium dan Appium, di mana dapat mendukung berbagai sistem operasi yang ada. Fungsionalitas yang komprehensif dalam mengimplementasikan pengujian otomatis untuk aplikasi web, API, desktop, dan seluler ditawarkan dalam Katalon Studio. Keuntungan menggunakan Katalon Studio adalah *freeware* atau tanpa biaya lisensi dan biaya pemeliharaan aplikasi, ini mengintegrasikan fitur-fitur yang diperlukan untuk membuat dan mengeksekusi *Test Case*, dan sangat mudah digunakan bahkan untuk pemula sekalipun[24]. Membuat *Test Case* di Katalon Studio dapat dilakukan dengan beberapa cara yaitu recording dan playback, mode manual dan mode script[25]. Katalon Studio memiliki keunggulan mudah digunakan, meminimalkan kesalahan manusia dan secara otomatis menghasilkan hasil pengujian.