

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian ini menggunakan sumber pustaka yang berhubungan dengan pengujian website yang menggunakan metode *Black box Testing*. Beberapa penelitian yang memiliki keterkaitan dengan penelitian yang dilakukan peneliti diantaranya adalah sebagai berikut:

Penelitian yang dilakukan oleh Ardi, F., & Putro, H. P. (2021) tentang pengujian Black Box Aplikasi Mobile Menggunakan Katalon Studio (Studi Kasus: ACC Partner PT. Astra Sedaya Finance). Hasil dari penelitian tersebut menunjukkan bahwa hasil pengujian automasi lebih efektif dibandingkan dengan pengujian secara manual yang memerlukan banyak waktu. *Tester* menjadi lebih mudah untuk mendapatkan *report* hasil pengujian karena terdapat Katalon *TestOps* yang menyajikan data hasil pengujian yang lebih detail dan dapat diunduh.

Penelitian yang dilakukan oleh Kosasih, Y., & Cahyono, A. B. (2021) Automation Testing Tool Dalam Pengujian Aplikasi *The Point Of Sale*. TPOS memiliki 3 aktor dan 2 situs. Aktor TPOS terdiri dari super admin, *use*, dan pegawai. Pengujian ini berfokus pada pengujian otomatis untuk mengembangkan aplikasi *The Point Of Sale* (TPOS) untuk mencari kecacatan dari sebuah aplikasi agar supaya nanti kalau sudah rilis cacatnya sudah sangat minim. Implementasi Katalon Studio terdiri dari beberapa tahapan, yaitu *test case*, *record & playback*

and result. Pengujian otomatis menggunakan Katalon Studio cukup efektif dari segi waktu dibandingkan pengujian manual.

Penelitian yang dilakukan oleh Putri, Y. F. (2020) *Automation Regression Testing Pada Aplikasi Teman Diabetes Dengan Menggunakan Metode Black Box Testing*. Fokus penelitian ini adalah bidang usaha GueSehat dengan produk yang diteliti adalah aplikasi Teman Diabetes, karena aplikasi ini masih memiliki banyak *bug* akibat adanya *human error* yang disebabkan dari durasi waktu yang lama dalam proses pengujian. Dalam penelitian ini, didapatkan hasil yaitu sebuah rekomendasi alur kerja pengujian regresi yang ter-otomatisasi menggunakan metode *continues integration* dengan *tool* Gitlab CI yang mengikuti *software testing* standard ISO / IEC 29119 dan dalam penelitian ini membuktikan bahwa pengujian regresi yang dilakukan dengan menentukan skala prioritas untuk *test case* yang akan diotomatisasikan dan membandingkan waktu pengujian regresi yang terotomatisasi lebih membutuhkan waktu yang singkat dibandingkan dengan pengujian regresi manual. Hasil yang didapatkan oleh peneliti dengan menggunakan metode *boundary value analysis* adalah semua hasil sudah lolos untuk *input* tanggal jatuh tempo, kota bank, dan cabang bank. Namun terdapat *bug* yang ditemukan pada validasi *input* nomor rekening dan pemilik rekening. Sedangkan hasil yang didapatkan oleh peneliti dengan menggunakan *decision table testing* adalah semua form yang diuji sudah lolos uji.

Penelitian yang dilakukan oleh Wahyuji, A. T. (2021) *Pengembangan dan Pengujian Aplikasi Website Career Center ITERA*. Hasil implementasi dan pengujian aplikasi *website career center* ITERA, penelitian ini berhasil merancang,

mengembangkan dan membangun aplikasi *website career center* ITERA dan aplikasi yang dikembangkan dapat membantu pengguna dalam mencari informasi lowongan kerja yang sesuai kebutuhan mereka dan memudahkan mereka untuk langsung mendaftarkan dirinya. Pengujian menggunakan teknik *white box* dengan *tool* Katalon Studio, objek yang diujikan sesuai dengan rancangan aktivitas aplikasi. Aplikasi web career center ITERA ketika diuji membutuhkan response time total 159.4 detik atau rata – rata 14.5 detik dengan hasil yang baik, hanya terdapat 1 kegagalan yaitu tidak bisa menerima masukkan tanggal sehingga aplikasi terdapat kesalahan. Katalon Studio tidak dapat mendeteksi masukkan berupa *datepicker* pada aplikasi web.

Penelitian yang kelima peneliti temukan Zulianto, A., Purbasari, A., Suryani, N., Susanti, A. I., Rinawan, F. R., & Purnama, W. G (2021) Pemanfaatan Katalon Studio untuk Otomatisasi Pengujian *Black-Box* pada Aplikasi iPosyandu. Hasil pengujian dari pengujian manual dan pengujian otomatis akan dibandingkan dengan parameter: status hasil eksekusi dan waktu eksekusi. Secara hasil eksekusi, pengujian manual dan otomatis menghasilkan status yang sama: 10pass dan 3 fail. Pada saat menjalankan pengujian dapat diketahui *respon time* setiap *test case*, *respon time* pada satu *test case* berbeda – beda apabila dijalankan beberapa kali. Total waktu eksekusi pengujian otomatis total adalah 238,08 detik, dan 719,27 detik untuk pengujian secara manual. Dengan demikian terjadi peningkatan kecepatan 2,54 kali atau 2,54%.

Penelitian yang keenam adalah penelitian yang diusulkan oleh Laurenzia Rika Dewi (2023) tentang perbandingan hasil *automation testing* Robot Framework

dan Katalon Studio pada e-learning SD Negeri 1 Dibal Boyolali. Dari penelitian ini didapatkan hasil pengujian Robot Framework dan Katalon Studio akan dibandingkan dengan parameter: status hasil eksekusi dan waktu eksekusi. Secara hasil eksekusi, pengujian robot framework dan pengujian katalon studio menghasilkan status yang sama: 57 berhasil 2 gagal. Total waktu eksekusi pengujian robot framework adalah 1 Jam 3 Menit 5 Detik dan 53 Menit 53 Detik untuk pengujian katalon studio. Dengan demikian selisih waktu pengujian 9 Menit 12 Detik.

Ringkasan refrensi penelitian yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Ringkasan Refrensi Penelitian

Penulis	Objek	Tools
Ardi, F., & Putro, H. P. (2021)	Aplikasi Mobile Menggunakan Katalon Studio ACC Partner PT. Astra Sedaya Finance.	Katalon Studio
Kosasih, Y., & Cahyono, A. B. (2021)	Aplikasi <i>The Point Of Sale</i>	Katalon Studio
Putri, Y. F. (2020)	Aplikasi Teman Diabetes	Katalon Studio
Wahyuji, A. T. (2021)	Aplikasi Website Career Center ITERA	Katalon Studio
Zulianto, A., Purbasari, A., Suryani, N., Susanti, A. I., Rinawan, F. R., & Purnama, W. G (2021).	Aplikasi iPosyandu	Katalon Studio
Laurenzia Rika Dewi (2023)	E-learning SD Negeri 1 Dibal Boyolali	Robot Framework dan Katalon Studio

2.2 Dasar Teori

2.2.1 Software Quality Assurance

Quality Assurance atau biasa disebut QA adalah seseorang yang bertanggung jawab terhadap perencanaan jaminan kualitas, kesalahan, penyimpanan rekaman, analisis dan pelaporan.

Jaminan kualitas perangkat lunak adalah aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak. Proses ini dilaksanakan oleh seorang *QA Tester*. *QA Tester* memiliki tugas utama yaitu melakukan pengujian terhadap perangkat atau emulator, membuat alur pengujian, serta membuat laporan hasil pengujian yang terkadang bertugas untuk membuat program pengujian otomatis, membuat laporan pengujian, memberikan masukan atas aplikasi yang diuji, serta berkomunikasi dengan pihak – pihak yang berkepentingan, seperti pengembang UI/UX, *Back-End* atau Project Manager (PM) (Zakiyah, 2018). Tujuan utama dari proses SQA adalah memastikan proses akhir dari sebuah produk berkualitas tinggi sesuai dengan persyaratan yang dinyatakan dalam standar yang telah ditetapkan. Sebuah perusahaan pengembang perangkat lunak mempunyai banyak proyek yang dikerjakan bersamaan, dan setiap pekerjaan memiliki rekrutment khusus dan tingkat kesulitan yang berbeda, oleh karena untuk perusahaan pengembang perangkat lunak harus memiliki standarisasi kustom terhadap perangkat lunak untuk menghasilkan hasil yang baik dan kualitas yang dapat diterima seperti yang diungkapkan (Zope et al (2015)).

Hariyanto, H., & Dirgahayu, T. (2020) menyatakan bahwa Kegagalan proyek pengembangan *software* dapat diakibatkan karena tidak adanya proses yang

menjamin kualitas *software* sehingga dapat meminimalisir risiko – risiko yang ada pada setiap tahapan pengembangan *software*. Risiko ini bersifat teknis dan non teknis. Risiko teknis misalnya adalah *software* tidak dapat berfungsi sebagaimana mestinya, atau sulit untuk dioperasikan, dimodifikasi atau dikelola. Adapun risiko non teknis misalnya adalah proyek melebihi biaya yang dianggarkan, atau jadwal melewati batas yang direncanakan. Untuk mengurangi risiko – risiko tersebut diperlukan serangkaian aktivitas yang memastikan bahwa setiap proses dilakukan sesuai standar. *Software Quality Assurance* (SQA) merupakan serangkaian kegiatan yang terencana dan sistematis yang memastikan bahwa proses dan produk *software* sesuai dengan persyaratan, standar, dan prosedur. Pengembangan *software* yang benar menentukan kualitas sebuah *software* yang dikembangkan. Dengan adanya aktivitas SQA ini diharapkan kebutuhan dan target *software* yang dikembangkan dapat tercapai. Selain itu, proses pengembangan *software* juga dapat dibuktikan bahwa proses pengembangan telah dilakukan dengan baik karena SQA mendokumentasikan setiap prosesnya dengan standar.

2.2.2 Robot Framework

Robot Framework atau Kerangka Robot adalah kerangka otomatisasi open source generik. Ini dapat digunakan untuk otomatisasi pengujian dan otomatisasi proses robot (RPA).

Robot Framework didukung oleh Robot Framework Foundation. Banyak perusahaan terkemuka di industri menggunakan alat ini dalam pengembangan perangkat lunak mereka. Kerangka robot terbuka dan dapat diperluas, kerangka robot dapat diintegrasikan dengan hampir semua alat lain untuk menciptakan solusi

otomatisasi yang kuat dan fleksibel. Kerangka robot gratis untuk digunakan tanpa biaya lisensi.

Kerangka robot memiliki sintaks yang mudah, menggunakan kata kunci yang dapat dibaca manusia. Kemampuannya dapat diperluas dengan perpustakaan yang diimplementasikan dengan Python, Java atau banyak bahasa pemrograman lainnya. Kerangka robot memiliki ekosistem yang kaya di sekitarnya, terdiri dari perpustakaan dan alat yang dikembangkan sebagai proyek terpisah (Robot Framework (2019)).

2.2.3 Katalon Studio

Katalon Studio adalah aplikasi *open source* untuk pengujian secara otomatis yang dikembangkan oleh Katalon LLC. Katalon mendukung 3 *platform* pengujian secara otomatis, yaitu *Web testing*, *API testing*, dan *Mobile testing*. Katalon sudah terintegrasi dengan beberapa teknologi luar seperti qtest, JIRA, kobiton, github, dan lain – lain (Katalon LLC (2020)).

Kebutuhan proses pengujian dilakukan dengan menggunakan *tool* yaitu Katalon Studio. *Tool* tersebut merupakan aplikasi open source untuk melakukan pengujian secara otomatis seperti Windows, MAC os dan linux. Katalon Studio menggunakan antarmuka IDE khusus untuk melakukan pengujian. Katalon Studio memiliki tiga fitur utama untuk menunjang beberapa pengujian di beberapa *platform* seperti *Web testing*, *API testing*, dan *Mobile testing* (Studio, 2020).

2.2.4 E-learning SD Negeri 1 Dibal Boyolali

E-learning SD Negeri 1 Dibal Boyolali dirancang sebagai sistem pembelajaran elektronik sebagai bentuk teknologi informasi yang ditetapkan

dibidang Pendidikan berupa website yang dapat diakses dimana saja. *E-learning* SD Negeri 1 Dibal Boyolali ini memiliki 3 user untuk mengakses *e-learning* yaitu User Siswa, User Guru, User Admin.

E-learning SD Negeri 1 Dibal Boyolali untuk user siswa memiliki 7 menu utama, yaitu *Login*, Tugas/Quiz, Nilai, Materi, Berita, Detail Profil, Edit Profil. E-learning SD Negeri 1 Dibal Boyolali untuk user guru memiliki 6 menu utama yaitu *Login*, Manajemen Kelas, Mata Pelajaran, Manajemen Tugas/Quiz, Materi, Berita. Dan e-learning SD Negeri 1 Dibal Boyolali untuk user admin memiliki 9 menu utama yaitu *Login*, Manajemen Pengajar, Manajemen Siswa, Manajemen Registrasi Siswa, Manajemen Kelas, Mata Pelajaran, Manajemen Tugas/Quiz, Materi, dan Berita.

Saat ini *E-learning* dapat disampaikan melalui internet, tetapi di masa lalu *E-learning* hanya dapat disampaikan dengan menggunakan metode berbasis komputer seperti CD-ROM. *E-learning* juga merupakan model pembelajaran yang mencakup beragam media penyampaian bahan ajar atau konten melalui situs di internet dengan menggunakan multimedia (ragam media yang dapat menyampaikan pesan teks, grafik, audio, video, animasi secara terintegrasi), televisi interaktif, kelas virtual (pembelajaran yang dimediasi komputer dan internet secara *synchronous/real time* dengan guru/pengajar dan pembelajar tidak berada dalam sebuah tempat/ruangan yang sama).

E-learning juga dapat digunakan sebagai salah satu upaya dalam meningkatkan pengetahuan dan kinerja melalui penggunaan teknologi internet. *E-learning* juga disebut sebagai pembelajaran berbasis web, pembelajaran dalam

jaringan (online), pembelajaran yang terdistribusi dengan instruksi yang dibantu oleh komputer atau disebut juga sebagai pembelajaran berbasis internet (Muhammad Rusli, M. T., Hermawan, D., & Supuwiningsih, N. N. ,2020))

2.2.5 Pengujian

Pegujian merupakan suatu rangkaian aktivitas yang terencana dan sistematis untuk menguji atau mengevaluasi kebenaran fungsi aplikasi berdasarkan desain kasus uji (test case) yang spesifik (Yulistina, Nurmala, Supriawan, Juni, & Saifudin, 2020). Pengujian akan direncanakan dengan tahapan awal yaitu menentukan *Test Case* yang akan diuji, lalu membaginya menjadi beberapa partisi masukan dan keluaran. Hal ini dilakukan buat menerima dokumentasi pengujian (MZ, 2016). Kemudian hasil *Test Case* akan dijadikan sebuah tabel dokumentasi dari perangkat lunak yang dikembangkan sesuai dengan fungsinya.

Automation testing adalah aktivitas pengujian script test yang berjalan secara otomatis yang membandingkan fungsi yang diharapkan dengan hasil yang sebenarnya. Pengujian dengan *automation testing* ini dapat dilakukan secara berulang sehingga apabila hasilnya tidak sesuai dengan yang diharapkan maka terdapat *bug* didalam program yang sedang dibangun. *automation testing* memungkinkan untuk melakukan *testing* secara berulang kali dalam waktu singkat sehingga dapat meningkatkan jangkauan dalam pengujian. (Putri, Y. F. (2020)).

Automation Testing atau pengujian otomatis memainkan peran penting dalam sistem SDLC arena menguji dengan cepat, di mana pengujian otomatis mengujakan *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 (Klammer, C., & Ramler, R.(2017)). Ini menjelaskan bahwa *automation testing* dapat disebut pengujian yang dinamis. Efektifitas 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. *Autination tools* yang sering digunakan adalah Watir, *Unified Functional Testing*, Katalon Studio, *Testcomplete*, dan selenium (Oliinyk, B., & Oleksiuk, V.(2019)).

2.2.6 Blackbox Testing

Black box testing didefinisikan sebagai metode pengujian tanpa melihat struktur kode internal, rincian implementasi, dan pengetahuan tentang jalur internal perangkat lunak. Jenis pengujian ini didasarkan sepenuhnya pada persyaratan dan spesifikasi perangkat lunak. Dalam *black box testing* hanya berfokus pada *input* dan *output* dari sistem perangkat lunak tanpa peduli tentang pengetahuan internal dari program perangkat lunak.

Menurut (Roni Setiawan, 2021) *Black box testing* atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat

lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, peguji tidak harus memiliki kemampuan menulis kode program. (Setiawan, 2021).