

Pengujian Website EPOS PT XYZ Menggunakan Metode *Black Box Testing*

Wendy Winata¹, Andi Wahyu Rahardjo Emanuel², Herlina³

Program Studi Informatika Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta
Jl. Babarsari 43, Kabupaten Sleman 55281, Daerah Istimewa Yogyakarta, Indonesia
Email: ¹wendywinata128@gmail.com, ²andi.emmanuel@uajy.ac.id, ³herlina@uajy.ac.id

Abstrak. PT XYZ merupakan salah satu perusahaan yang mengadopsi website untuk membantu produktivitas bisnisnya. Oleh karena itu, website yang digunakan memerlukan pengujian terlebih dahulu untuk memastikan fungsionalitas website dapat berjalan sesuai dengan tujuan dan terhindar dari bug atau masalah. Penelitian ini berfokus dalam melakukan pengujian pada website secara manual dan otomatis, menggunakan metode black box testing dan teknik equivalence partitioning. Pengujian akan dimulai dengan membangun test case terlebih dahulu agar bisa mendapatkan hasil yang terstruktur dan terarah. Hasil dari pengujian akan digunakan untuk melihat bagaimana presentase keberhasilan dari fungsionalitas yang ada pada website. Selain mendapatkan presentase keberhasilan dari pengujian, penelitian ini juga akan melihat bagaimana kelebihan dan kekurangan dari pengujian otomatis dan pengujian manual berdasarkan beberapa parameter perbandingan.

Kata Kunci: Pengujian, Black Box Testing, Equivalence Partitioning, Pengujian Otomatis, Pengujian Manual

1. Pendahuluan

1.1. Latar Belakang

Website sebagai media informasi merupakan salah satu teknologi yang ada di kehidupan manusia. Perkembangan website telah sangat mempermudah aktivitas manusia seperti mencari informasi, menyimpan informasi, hingga berkomunikasi. Kemudahan pengaksesan website membuat perkembangan informasi menjadi lebih efisien dan *up to date* [1]. Dengan keunggulan dan kemudahan yang ditawarkan, website menjelma menjadi salah satu aktivitas yang di gemari oleh manusia. Bahkan menurut Kominfo, pengguna internet di Indonesia mencapai 175,5 juta orang pada tahun 2020 [2].

Hal ini membuat banyak perusahaan mulai memanfaatkan website untuk membantu produktivitas dari pekerjaan. Pemanfaatan ini dimulai dari membuat website perusahaan, membuat sistem informasi untuk perusahaannya, hingga mengubah aktivitas yang dilakukan *manual* menjadi aktivitas yang dilakukan secara *online*. Perkembangan ini juga menyebabkan kebutuhan baru bagi sebuah perusahaan yaitu membuat sebuah website atau aplikasi yang dapat menyelesaikan permasalahan dari perusahaan.

Meningkatnya kebutuhan perusahaan akan aplikasi menyebabkan aktivitas perusahaan akan memiliki ketergantungan pada aplikasi. Karena ketergantungan tersebut, perusahaan perlu memastikan kualitas dari aplikasi yang dibuat dapat memiliki kualitas yang baik dari segi fungsionalitas, hingga tampilan aplikasi agar tidak menghambat aktivitas perusahaan.

Kualitas dari sebuah aplikasi dapat dinilai dari banyak hal, diantaranya adalah fungsionalitas yang sesuai dengan kebutuhan pengguna, *usability* yang memastikan aplikasi mudah dioperasikan dan dipelajari, efisiensi kinerja aplikasi yang tidak boros, kehandalan aplikasi untuk mempertahankan kinerjanya, *maintainability* suatu aplikasi untuk dapat dikembangkan sesuai dengan perubahan lingkungan, dan portabilitas yang merupakan kemampuan aplikasi untuk beradaptasi saat di gunakan di area tertentu [3]. Namun sayangnya, terdapat beberapa aplikasi yang memiliki masalah saat aplikasi tersebut muncul di publik, masalah ini dapat menyebabkan buruknya *user experience* bagi penggunanya. Buruknya *user experience* bagi pengguna dapat menyebabkan pengguna meninggalkan aplikasi dan beralih ke aplikasi lain. Masalah yang terjadi pada aplikasi, juga dapat membuat terhambatnya pekerjaan yang harus dilakukan karena terjadi masalah pada aplikasi tersebut.

Dalam rangka mengatasi permasalahan yang mungkin terjadi pada sebuah aplikasi, perusahaan perlu melakukan pengujian aplikasi terlebih dahulu untuk memastikan aplikasi yang dibuat layak

muncul di publik. Pengujian aplikasi merupakan proses untuk mencari kesalahan atau *bug* pada suatu aplikasi yang disertai dengan evaluasi terhadap seluruh fungsi untuk memastikan aplikasi yang dibuat telah sesuai dengan tujuannya atau masih memiliki kecacatan. Dengan adanya pengujian, perusahaan dapat meminimalisir terjadinya sebuah *bug* dan memastikan seluruh fungsionalitas dapat berjalan dengan baik. Salah satu metode pengujian yang cukup populer adalah *black box testing*. Pengujian *black box testing* berfokus pada fungsionalitas aplikasi tanpa perlu mengetahui bagaimana aplikasi tersebut dibentuk atau dibuat, sehingga pengujian dapat lebih mudah dilakukan.

PT XYZ merupakan salah satu perusahaan yang berusaha mengadopsi teknologi untuk membantu produktivitas dari bisnis asuransi yang ada pada perusahaan. PT XYZ berencana membuat sebuah aplikasi berbasis website bagi agennya agar dapat membantu produktivitas dalam melakukan pengkinian data nasabah, pembuatan ilustrasi asuransi, hingga pengajuan asuransi bagi calon nasabahnya. Dengan adanya website ini, nasabah dapat mengetahui bagaimana pembayaran yang dilakukan oleh nasabah hingga manfaat dari produk asuransi yang dipilih melalui proposal yang telah dibuat.

Dalam mengembangkan aplikasi berbasis website ini, maka perlu dilakukan pengujian pada aplikasi tersebut untuk memastikan fungsionalitas aplikasi dapat berjalan dengan lancar dan terhindar dari *bug* yang mungkin terjadi ketika aplikasi telah memasuki fase production. Pada penelitian ini, pengujian akan dilakukan dengan berfokus pada fungsionalitas sistem. Pengujian akan menerapkan metode *black box testing* dengan teknik *equivalence partitioning*. Tujuan dari *black box testing* adalah mengetahui error atau *bug* yang terjadi sedini mungkin [4]. Metode ini juga sangat baik digunakan untuk menemukan kesalahan seperti kesalahan validasi data, output data yang tidak sesuai, dan kesalahan performa [5]. Pengujian akan dilakukan secara *manual* dan otomatis. Pada *manual*, pengujian akan langsung dilakukan setelah dilakukan pembangunan *test case*. Pada otomatis, pengujian di mulai dengan melakukan pengkodean atau membangun sebuah *script* (skenario) dengan menggunakan bahasa Javascript. *Script* yang telah dibuat akan dieksekusi menggunakan teknologi Selenium. Selenium sendiri merupakan teknologi *framework* pada bahasa Javascript yang dapat digunakan untuk membantu pengujian otomatis pada sebuah website.

Berdasarkan permasalahan yang telah dijabarkan pada latar belakang, maka diperoleh rumusan masalah yaitu: (1) Apakah fungsionalitas yang di uji pada website EPOS telah berjalan dengan baik dan berhasil? (2) Bagaimanakah keunggulan dan kekurangan dari masing masing pengujian yang telah dilakukan? Penelitian yang dilakukan berdasarkan rumusan masalah yang telah dipaparkan sebelumnya akan memiliki beberapa batasan yaitu: (1) Website di uji menggunakan metode *black box testing*, dengan teknik *equivalence partitioning*. (2) Pengujian dilakukan secara *manual* dan otomatis dengan *framework* Selenium. (3) Pengujian dilakukan pada Website EPOS terbatas pada pembaruan terakhir Mei 2022. (4) Pengujian pada website EPOS dilakukan hanya terbatas dari sisi Agen. Penelitian ini akan memiliki tujuan sebagai berikut: (1) Mengetahui apakah fungsionalitas yang di uji pada website EPOS telah berjalan sesuai dengan kondisi yang diharapkan. (2) Mengetahui keunggulan dan kekurangan dari pengujian otomatis dan pengujian *manual*.

2. Tinjauan Pustaka

Pengujian merupakan salah satu bagian dari SLDC yang sangat penting karena berperan untuk meningkatkan kualitas suatu aplikasi. Pengujian juga akan meminimalisir terjadinya masalah ketika aplikasi telah *dideploy* ke publik. Bab ini menjabarkan beberapa penelitian terdahulu yang memiliki kaitan dengan pengujian aplikasi khususnya dalam metode *black box testing*.

Pengamatan pertama merupakan pengujian yang dilakukan oleh Vanessa Angela Amei dengan objek yang menjadi pengujian merupakan sebuah aplikasi yang dimiliki oleh perusahaan Astra Credit Companies yaitu ACC Revamp. Dalam pengujian, peneliti melakukan dua macam pengujian yaitu, pengujian *manual* dan pengujian otomatis menggunakan Katalon Studio. Penelitian dimulai terlebih dengan melakukan analisis dokumen, dan pembangunan *test case*. Setelah *test case* berhasil dibangun, pengujian secara *manual* akan dimulai dengan memasukan input bernilai *valid* dan *invalid* dengan memperhatikan hasil keluaran dari pengujian yang

dilakukan. Selanjutnya, peneliti melakukan pengujian otomatis dengan membangun *script* dengan metode *record and replay* dengan Katalon Studio. Pada penelitian ini, peneliti mendapatkan hasil bahwa pengujian *manual* berjalan lebih cepat dibandingkan pengujian otomatis dengan rata-rata 62%. Selain itu, peneliti juga mendapatkan dua *bug* dari lima belas fungsi yang diuji [6].

Pengamatan selanjutnya dilakukan pada pengujian aplikasi Doctor to Doctor oleh Albert Teonando Suhardi. Dalam pengujian ini, peneliti akan melakukan pengujian secara *manual* dan otomatis menggunakan metode *equivalence partitioning* dengan bantuan tools Katalon. Setelah pengujian selesai dilakukan, peneliti membandingkan kedua pengujian dan mendapatkan pengujian *manual* lebih efektif dari pada pengujian otomatis jika ditinjau dari waktu untuk menjalankan seluruh fungsi dengan selisih waktu sebanyak 10 menit 55 detik. Selain itu presentase keberhasilan fungsional aplikasi juga cukup tinggi yaitu 86,89%, dan dikategorikan sebagai sangat baik [7].

Pengamatan ketiga dilakukan pada pengujian otomatis pada aplikasi ACC Leads dengan metode *black box testing* oleh Megawati Theresia. Dalam pengujian, peneliti menggunakan teknik *boundary value analysis* dan *equivalence partitioning* dengan scope yang diuji adalah fitur yang bisa menerima masukan (baik berupa angka maupun huruf) pada aplikasi mobile ACC Leads. Setelah pengujian dilakukan, peneliti juga kembali akan melakukan evaluasi terhadap hasil pengujian tersebut. Berdasarkan pengujian yang telah dilakukan, peneliti menyimpulkan aplikasi ACC Leads masih belum mampu memberikan *feedback* ketika diberi inputan yang tidak *valid* seperti pada saat pengguna memberikan inputan yang jumlah karakternya sudah melebihi limit [8].

Pengamatan selanjutnya dilakukan pada pengujian sistem informasi pernikahan *online* berbasis website dengan metode *black box testing* yang dilakukan oleh Umi Salamah dan Fata Nidaul Khasanah. Pada pengujian *manual*, peneliti akan membuat tabel *test case* terlebih dahulu untuk menentukan fungsionalitas yang ingin diuji, setelah itu peneliti langsung mencoba menjalankan fungsionalitas website secara *manual* berdasarkan *test case* yang telah dibuat sebelumnya. Setelah pengujian selesai, peneliti menyimpan seluruh hasil pengujianya untuk membandingkannya dengan hasil yang diharapkan sesuai dengan *test case* yang telah dibuat. Pengujian kedua dilakukan dengan menyebar kuisioner yang berisi penilaian terakit *user interface* dan *user experience* pada aplikasi pernikahan *online*, peneliti memberi empat buah pilihan dari setiap kriteria yang ditanyakan, empat pilihan tersebut adalah tidak baik, kurang baik, baik, dan sangat baik. Berdasarkan kedua pengujian yang telah dilakukan, peneliti mendapatkan hasil bahwa seluruh fungsionalitas sistem telah berjalan sesuai dengan harapan, selain itu berdasarkan kuisioner yang telah disebar, aplikasi pernikahan *online* mendapatkan rata-rata nilai 81% dari segi *user interface* dan *user experience* sehingga tingkat kepuasan pengguna pada website dapat dikategorikan tinggi [9].

Pengamatan terakhir merupakan penelitian tentang pengujian aplikasi menggunakan metode *black box testing* dengan teknik *boundary value analysis* yang diteliti oleh M. Sidi Mustaqbal, dkk. Pada pengujian ini, objek dari pengujian merupakan aplikasi prediksi kelulusan SNMPTN. Pengujian ini akan berfokus pada beberapa fungsionalitas aplikasi. Peneliti memulai pengujian dengan menyiapkan terlebih dahulu data uji, lalu mendefinisikan skenario dari form yang ingin diuji. Dengan menggunakan *boundary value analysis*, peneliti menyimpan hasil pengujian aplikasi dalam tabel yang disebut tabel hasil uji yang memiliki empat buah attribute yaitu masukan, hasil yang diharapkan, hasil sebenarnya, dan kesimpulan. Berdasarkan tabel hasil uji yang telah dibuat, peneliti menyimpulkan bahwa aplikasi masih memiliki beberapa kekurangan yaitu belum lengkapnya proses validasi data pada aplikasi sehingga dibutuhkan penyempurnaan fungsi validasi pada aplikasi [10].

3. Metodologi Penelitian

Metodologi yang dilakukan pada penelitian dan pengujian aplikasi EPOS adalah sebagai berikut: (1) Studi Pustaka, tahap penelitian dimulai dari mencari referensi penelitian yang pernah dilakukan sebelumnya untuk memperkuat informasi dalam melakukan pengujian. (2) Analisis Skenario Aplikasi, pada tahap ini peneliti akan mempelajari skenario dan kebutuhan

aplikasi melalui dokumen *user guide* dan *functional system design* untuk dapat mengetahui lebih dalam tentang kebutuhan dan cara kerja aplikasi. (3) Merancang *Test Case* Skenario Pengujian, tahapan selanjutnya adalah membuat *test case* yang nantinya akan berfungsi sebagai validator dari pengujian yang akan dilakukan. *Test case* juga akan membantu pengujian berjalan secara terarah dan efektif. (4) Melakukan Pengujian *Manual*, tahapan ini merupakan implementasi pengujian yang pertama, pengujian akan dilakukan secara *manual* menggunakan *browser* berdasarkan *test case* yang dibuat sebelumnya. (5) Melakukan Pengujian Otomatis, tahapan ini merupakan implementasi pengujian yang kedua, pengujian akan dilakukan dengan membangun *script* yang akan dieksekusi otomatis oleh Node JS. (6) Penulisan Laporan Akhir, tahapan terakhir merupakan penulisan laporan dan kesimpulan dari pengujian yang telah dilakukan.

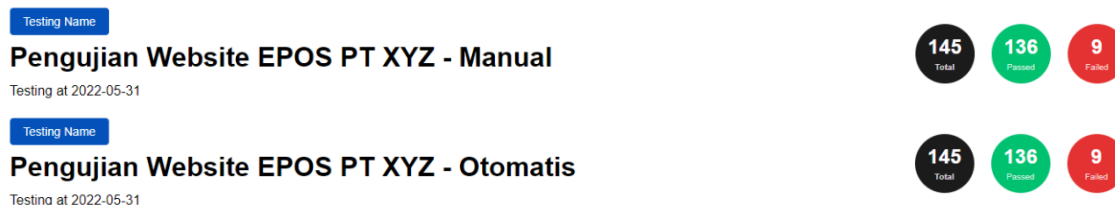
4. Hasil dan Diskusi

4.1 Perancangan *Test Case*

Pengujian bertujuan untuk mengetahui bagaimana ketepatan dan fungsionalitas yang dimiliki oleh suatu aplikasi atau website. Dalam melakukan pengujian, peneliti akan menggunakan *test case* yang dirancang untuk menjadi dasar dalam melakukan pengujian. Pembuatan *test case* bertujuan agar pengujian yang dilakukan memiliki rancangan yang terstruktur terhadap tindakan yang akan dilakukan oleh *user*. *Test case* juga akan berperan sebagai validasi dan verifikasi dari skenario yang dilakukan berdasarkan output (*actual result*) yang terjadi ketika pengujian dilakukan, dengan *expected result* yang di dapat dari dokumen *user requirement*. *Test case* akan dibangun dengan atribut sebagai berikut : (1) Nama dan Deskripsi Use Case. (2) Aktor Use Case. (3) Pre Conditions. (4) Post Conditions. (5) Basic Flow. (6) Alternative Flow. (7) Matriks Skenario. (8) Identifikasi Variabel. (9) *Validity Check*. (10) Matriks *Test Case*.

Dalam merancang *test case*, peneliti akan menggunakan teknik *equivalence partitioning*. Peneliti akan membagi domain dari suatu inputan atau *field* menjadi dua domain yaitu *valid* dan tidak *valid*. Penentuan *valid* dan tidak *valid* dari suatu *field* akan didasarkan oleh dokumen *user requirement* atau *functional system design* yang telah dipelajari sebelumnya. Lalu, berdasarkan domain yang telah dibagi, peneliti akan menyusun matriks *test case* yang terdiri dari *test case* positif (dalam range *valid*) dan *test case* negatif (dalam range *invalid*), tujuan dari pembagian ini adalah untuk melihat bagaimana fungsionalitas aplikasi dalam menjalankan sebuah fungsi dan menangani inputan *invalid* atau skenario negatif. Perancangan *test case* yang dilakukan pada 25 scenario yang akan diuji menghasilkan 145 *test case*.

4.2 Hasil Pengujian



Gambar 1. Hasil Pengujian *Manual* dan Otomatis

Pengujian *manual* berhasil diselesaikan dalam waktu 66 menit lebih 10 detik dan pengujian otomatis berhasil diselesaikan dalam waktu 39 menit lebih 4 detik. Kedua pengujian yang dilakukan dengan menggunakan 25 scenario dan total 145 *test case* memiliki hasil yang sama yaitu menghasilkan 136 test berhasil dan 9 test gagal dengan presentase keberhasilan sebesar 93,80%. Presentase ini menunjukkan website EPOS memiliki tingkat keberhasilan yang tinggi, namun sayangnya aplikasi masih memiliki kekurangan dalam menangani beberapa skenario negatif.

Terdapat beberapa skenario dimana aplikasi EPOS tidak dapat melakukan validasi atau menampilkan warning sesuai dengan inputan *invalid* yang diberikan. Hal ini tentunya dapat

menyebabkan kebingungan bagi user yang berdampak terhambatnya fungsionalitas aplikasi. Skenario yang gagal atau tidak dapat ditangani diantaranya adalah : (1) Skenario Ganti Password, pada skenario ini terdapat tiga test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*, pengecekan perbedaan konfirmasi password dengan password baru, dan pengecekan password lama yang salah. (2) Skenario Tambah Kontak, pada skenario ini terdapat dua test yang gagal berfungsi dengan baik, website gagal dalam melakukan validasi pada inputan nomor hp, sehingga nomor hp yang masuk ke dalam sistem tidak sesuai dengan ketentuan *field* tersebut. (3) Skenario Penambahan Ilustrasi, pada skenario ini terdapat satu test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*. (4) Skenario Penambahan SPAJ Step 1, pada skenario ini terdapat satu test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*. (5) Skenario Penambahan SPAJ Step 2, pada skenario ini terdapat satu test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*. (6) Skenario Penambahan SPAJ Step 3, Pada skenario ini terdapat satu test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*. (7) Skenario penambahan SPAJ step 10, pada skenario ini terdapat satu test yang gagal berfungsi dengan baik, website gagal dalam memunculkan warning ketika pengujian menjalankan skenario negatif pada pengecekan *field mandatory*.

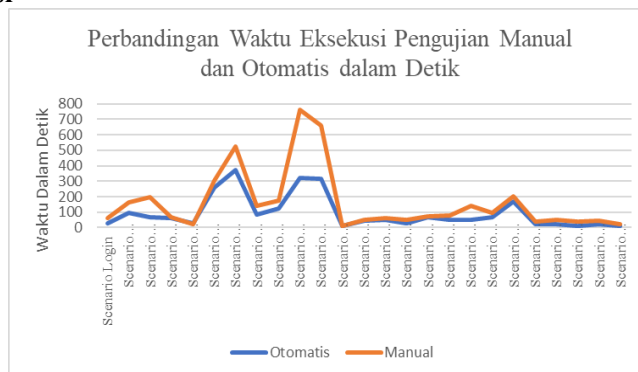
4.3 Perbandingan Pengujian

Pengujian otomatis dan pengujian *manual* yang telah peneliti laksanakan berhasil mendapat presentase keberhasilan yang sama. Hal ini menunjukkan kedua pengujian sama sama baik untuk menguji sebuah aplikasi. Pada bagian ini, peneliti akan menjabarkan keunggulan dan kekurangan dari masing masing pengujian berdasarkan variable tertentu, perbandingan tersebut adalah :

4.3.1. Persiapan Pengujian

Pengujian otomatis memerlukan persiapan yang lebih banyak dan lebih lama daripada pengujian manual. Penguji harus mempelajari dokumen dan website terkait, lalu mempersiapkan alat yang dapat digunakan untuk membuat sebuah *script* dan mengeksekusi *script* tersebut. Dalam hal ini, peneliti perlu mempersiapkan penginstalan aplikasi dan library yang dibutuhkan dalam pembuatan pengujian automated testing, misalnya saja untuk dapat membuat *script* dan mengeksekusinya, peneliti perlu menginstall Selenium sebagai framework web driver yang akan memfasilitasi interaksi dengan elemen di website, dan Node JS dan Javascript untuk dapat mengeksekusi kode yang ditulis dengan framework Selenium.

4.3.2. Waktu Eksekusi



Gambar 2. Grafik waktu pengujian otomatis dan manual

Pengujian *manual* berhasil diselesaikan dalam waktu 66 menit lebih 10 detik dan pengujian otomatis berhasil diselesaikan dalam waktu 39 menit lebih 4 detik, menunjukkan pengujian otomatis memiliki waktu eksekusi yang jauh lebih cepat daripada eksekusi *manual*, hal ini dikarenakan pengujian otomatis akan bergantung pada *script* yang telah dibuat dan pengujian otomatis di eksekusi oleh mesin (alat) sehingga hasil dari pengujian akan lebih cepat dan konsisten. Pengujian manual memiliki ketergantungan yang besar pada seorang penguji. Penguji dapat merasakan kelelahan, memiliki daya ketik yang lambat, atau kurang fokus dan mengakibatkan terjadinya human error dalam pengujian.

Pengujian otomatis juga memungkinkan sebuah *script* di eksekusi oleh mesin secara paralel tanpa ada batasan jumlah eksekusi dalam satu waktu (dapat di eksekusi pada *platform* operasi yang berbeda dan berkali kali dalam satu waktu), sedangkan untuk dapat mengeksekusi pengujian manual secara paralel dibutuhkan sumber daya manusia yang lebih banyak. Berdasarkan pengujian yang dilakukan oleh peneliti, peneliti mendapatkan pengujian otomatis website EPOS PT XYZ 1.7x lebih cepat daripada pengujian manual.

4.3.3. Sumber Daya

Pengujian otomatis membutuhkan sumber daya manusia dengan pengetahuan programming, hal ini dikarenakan pembuatan *script* akan menggunakan salah satu bahasa pemrograman dan kemampuan programming juga dibutuhkan untuk melakukan solving masalah saat pengcodingan. Programmer tersebut juga harus memiliki pengetahuan tentang framework atau perangkat lunak yang dipakai untuk melakukan pengujian (misalnya : Selenium, Appium, dan Katalon). Selain sumber daya manusia, dalam membuat *script* dan mengeksekusi pengujian, juga dibutuhkan perangkat keras yang dapat memenuhi kebutuhan tersebut (Laptop dengan spesifikasi tertentu).

Pengujian manual relatif membutuhkan sumber daya manusia yang lebih mudah dicari. Dalam melakukan pengujian manual, alat yang dibutuhkan tidak memerlukan spesifikasi yang berat, terkecuali aplikasi yang diuji membutuhkan spesifikasi tertentu.

4.3.4. Biaya

Pengujian manual memiliki biaya yang lebih murah daripada pengujian otomatis, hal ini dikarenakan lebih sedikitnya sumber daya yang dibutuhkan oleh pengujian manual. Pengujian otomatis akan membutuhkan lebih banyak sumber daya beserta perawatan untuk sumber daya tersebut, misalnya untuk meng-eksekusi suatu pengujian dibutuhkan laptop dengan spesifikasi tertentu dan perawatan tertentu.

4.3.5. Konsistensi

Pengujian manual memiliki ketergantungan pada manusia, hal ini menyebabkan tingkat keakuratan pengujian yang bisa tidak konsisten, Dengan mempertimbangkan beberapa variable pada manusia yang dapat mengarah kepada human error, seperti kelelahan, tidak fokus, kebosanan, dan lain lain dapat menyebabkan tingkat keakuratan pengujian manual bisa saja rendah. Sedangkan pengujian otomatis akan bergantung pada mesin yang tidak bisa lelah, sehingga hasil akan lebih konsisten dan memiliki tingkat keakuratan tinggi karena pasti akan dijalankan sesuai dengan *script* yang telah disusun.

4.3.6. Tingkat Pengamatan

Pengujian manual memungkinkan melakukan pengujian dengan pertimbangan manusia, hal ini dapat membantu menguji sebuah website dengan juga memperhatikan bagaimana UI yang ada pada website, sehingga website tersebut lebih ramah terhadap penggunaannya. Sedangkan pengujian otomatis akan bergantung pada *script* yang telah dibuat dan di eksekusi oleh mesin sehingga tidak bisa mempertimbangkan sesuatu.

4.3.7. Ketergantungan pada UI

Pengujian otomatis akan bergantung pada elemen yang ada pada website, sehingga ketika terjadi sedikit perubahan pada elemen di website misalnya penempatan elemen, id elemen, atau *class* elemen dapat membuat *automated testing* menjadi gagal dan perlu disesuaikan. Sedangkan pada pengujian manual, perubahan UI tidak akan menghalangi *tester manual* untuk dapat menjalankan pengujian.

4.3.8. Perekaman

Proses pengujian manual tidak dapat direkam, ketika ingin mengulangi suatu pengujian, seorang penguji harus melakukannya dari awal. Pada pengujian otomatis, *script* yang telah dibuat, dapat di panggil kembali, sehingga proses eksekusi dapat menjadi lebih mudah dan konsisten.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pengujian yang telah peneliti lakukan terhadap 25 skenario pada website EPOS PT XYZ menunjukkan presentase keberhasilan yang cukup tinggi yaitu 93,80% atau 136 test berhasil dari 145 *test case* yang ada. Tetapi website EPOS, masih memerlukan perbaikan khususnya dalam menangani skenario negative atau inputan yang tidak *valid*. Dengan terdapat beberapa inputan yang tidak memiliki warning, atau tidak memiliki validasi sesuai dengan FSD yang ada, dapat menyebabkan kebingungan bagi user saat tidak sengaja melakukan scenario negative.

Dilihat dari waktu eksekusi, pengujian otomatis memiliki waktu eksekusi yang lebih baik daripada pengujian *manual* dengan perbedaan hingga 1.7x lipat lebih cepat. Tetapi lamanya persiapan, besarnya biaya, dan kebutuhan sumber daya untuk *maintenance* dapat menjadi pertimbangan tertentu dalam menggunakan pengujian otomatis, sebaliknya cost dari pengujian *manual* jauh lebih murah dan persiapan yang lebih sebentar dapat menjadi opsi yang baik pula untuk melaksanakan pengujian.

Berdasarkan pengamatan peneliti, pengujian otomatis akan lebih optimal ketika dipakai pada projek dengan skala yang lebih besar, hal ini dikarenakan semua pengujian yang telah dilakukan sebelumnya, telah disimpan dalam *script* dan dapat dipanggil kapan saja, sedangkan pengujian *manual* akan lebih optimal pada projek dengan skala yang lebih kecil, karena waktu persiapan dan biaya yang dikeluarkan jauh lebih rendah daripada pengujian otomatis.

5.2 Saran

Aplikasi EPOS merupakan aplikasi yang saat ini masih akan dikembangkan untuk memenuhi kebutuhan seluruh produk asuransi yang ada di PT XYZ. Karena kebutuhan aplikasi yang masih akan dikembangkan, peneliti menyarankan untuk menggunakan pengujian otomatis dalam menguji aplikasi EPOS. Pengujian otomatis akan membantu pengujian seluruh fungsi dalam jangka panjang karena dapat dieksekusi berulang kali sesuai dengan pengembangan aplikasi. Pengujian otomatis juga dapat mempercepat sebuah pengujian karena *script* yang telah disimpan.

Pengujian yang dilakukan pada aplikasi EPOS telah memiliki presentase keberhasilan yang tinggi. Namun, aplikasi EPOS masih memiliki kekurangan khususnya dalam menangani kasus inputan yang tidak *valid* atau scenario negatif. Penulis menyarankan untuk lebih memperhatikan validasi dan warning yang ada pada inputan untuk memastikan data yang dikirim atau disimpan telah sesuai dengan FSD dan dapat diperiksa oleh departemen *underwriting*.

Referensi

- [1] P. S. Hasugian, "Perancangan website sebagai media promosi dan informasi," *J. Inform. Pelita Nusant.*, vol. 3, no. 1, pp. 82–86, 2018.

- [2] M. Fauziah, "Kominfo : Pengguna Internet di Indonesia Capai," *Republika.Co.Id*, pp. 2–3, 2021. [Online]. Available: <https://republika.co.id/berita/qhgibx335/kominfo-pengguna-internet-di-indonesia-capai-1755-juta-jiw#:~:text=REPUBLIKA.CO.ID%2C JAKARTA,populasi sebanyak 268.583.016 penduduk.>
- [3] Junyati, "Kualitas Software Model ISO 9126 – School of Information Systems," *Https://Sis.Binus.Ac.Id/*, 2019, [Online]. Available: <https://sis.binus.ac.id/2019/04/04/kualitas-software-model-iso-9126/>
- [4] A. Alfarizi, "Perancangan Aplikasi Profil Madrasah Di Jambi Luar Kota Berbasis Android(Studi Kasus : Pengawas Madrasah Kementerian Agama Kabupaten Muaro Jambi)," pp. 57–61, 2018, [Online]. Available: <http://repository.unama.ac.id/721/>
- [5] A. Agustian, I. Andryani, S. Khoerunisa, A. Pangestu, and A. Saifudin, "Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, p. 178, 2020, doi: 10.32493/jtsi.v3i3.5371.
- [6] V. A. Amei, "Pengujian Website Acc.Co.Id Revamp Menggunakan Metode Black Box Testing," 2021. [Online]. Available: <https://e-journal.uajy.ac.id/24407/>
- [7] A. T. Suhardi, "Pengujian Aplikasi Doctor to Doctor Menggunakan Metode Black Box Testing pada PT. Global Urban Essensial," (*Skripsi Unpubl. Univ. Atma Jaya Yogyakarta*, 2021.
- [8] M. Theresia, "AUTOMATION TESTING APLIKASI MOBILE ACC LEADS MENGGUNAKAN METODE BLACK BOX TESTING DENGAN TEKNIK BOUNDARY VALUE ANALYSIS DAN EQUIVALENCE PARTITIONING Tugas Akhir," 2021.
- [9] U. Salamah and F. Khasanah, "Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing," *Inf. Manag. Educ. Prof.*, vol. 2, no. 1, pp. 35–46, 2017.
- [10] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)," vol. I, no. 3, pp. 31–36, 2015.