

## Pengujian Website ACC.CO.ID Revamp Menggunakan Metode Black Box Testing

Vanessa Angela Amei<sup>1</sup>, Andi Wahyu Rahardjo Emanuel<sup>2</sup>, Stephanie Pamela Adithama<sup>3</sup>

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta Jl. Babarsari No. 43, Janti, Caturtunggal, Kec Depok, Kabupaten Sleman, 55281, Daerah Istimewa Yogyakarta, Indonesia

Email: <sup>1</sup>vanessangela11@gmail.com, <sup>2</sup>andi.emmanuel@uajy.ac.id, <sup>3</sup>stephanie.pamela@uajy.ac.id

**Abstrak.** *Acc.co.id Revamp adalah website yang berisi informasi mengenai perusahaan dan sebagai sarana online untuk memberikan layanan pembiayaan pada proses bisnis PT. Astra Credit Companies. Website Acc.co.id mengalami pengembangan ulang untuk dapat meningkatkan kualitas pelayanan dalam proses transaksi, sehingga pengujian terhadap website ini perlu dilakukan agar kualitas website sesuai dengan user requirement pengembangan awal. Pengujian yang dilakukan dalam penelitian ini menggunakan metode black box testing berbasis equivalence partitioning untuk menguji fungsionalitas website. Pengujian dilakukan dengan menerapkan dua teknik pengujian yaitu manual dan otomatis yang memanfaatkan tool Katalon Studio. Pengujian difokuskan pada 15 fungsi dalam website Acc.co.id Revamp. Berdasarkan hasil pengujian, terdapat perbedaan kinerja antara pengujian manual dan otomatis. Dari segi waktu, pengujian manual berjalan lebih cepat dibandingkan dengan teknik otomatis dengan persentase rata-rata waktu pengujian yaitu sebesar 55%. Selain itu, ditemukan adanya dua bug dari 15 fungsi yang diuji yang disebabkan karena terdapat ketidaksesuaian dengan user requirement yang telah disepakati.*

**Kata Kunci:** *Acc.co.id, Katalon Studio, black box testing, manual testing, equivalence partitioning*

### 1. Pendahuluan

Situs web adalah salah satu media *online* yang dapat menjadi wadah bagi pelaku bisnis untuk menciptakan cara-cara inovatif dalam perusahaan untuk melakukan bisnis dan berinteraksi dengan pelanggan [1]. Berdasarkan data yang dikumpulkan oleh organisasi We Are Social, jumlah pengguna internet di Indonesia meningkat 27 juta atau bertambah sebanyak 16% antara tahun 2020 dan 2021 [2]. Data tersebut menunjukkan bahwa informasi dan kualitas yang disediakan dalam *website* dapat menjadi suatu peranan penting dalam suatu proses bisnis perusahaan dan membentuk citra perusahaan di tengah-tengah masyarakat [3]. Hal inilah yang mendasari pentingnya untuk membangun *website* yang berkualitas dalam menjalankan proses bisnis bagi PT. Astra Credit Companies (ACC) yang bergerak dalam sektor pembiayaan mobil dan alat berat di Indonesia sejak tahun 1982.

Pengembangan ulang *website* Acc.co.id atau Acc.co. id Revamp adalah langkah yang diambil oleh pihak Astra Credit Companies (ACC) untuk meningkatkan kualitas pelayanannya kepada pengguna dengan membangun *website* dengan format yang lebih ringan dan mudah digunakan. Sebagai *website* yang menjadi penghubung utama antara ACC dan pelanggannya, Acc.co.id yang berfungsi untuk memberikan informasi kepada pengguna mengenai informasi perusahaan dan menyediakan fasilitas layanan untuk produk pembiayaan mulai dari mobil baru, mobil bekas, multiguna, properti, dan alat berat, diharapkan melalui proses pembaharuan *website* ini dapat menjawab semua kebutuhan penggunanya dalam melakukan proses layanan melalui pembangunan *website* dengan data leads yang terkoneksi, terdata, dan terbaharu. Oleh karena itu, pengujian terhadap *website* Acc.co.id Revamp perlu dilakukan untuk dapat memeriksa kualitas *website* agar dapat berjalan sesuai dengan *user requirement* pengembangan.

Pengujian akan difokuskan dengan batasan untuk menguji 15 fungsi yang ada di *website* Acc.co.id Revamp yang telah dipilih dengan pertimbangan resiko berapa sering fungsi dipakai,

keefisienan dalam penerapan *autotest* dan kemungkinan kesalahan terjadi pada setiap fungsi saat dijalankan. Pengujian akan menerapkan penggunaan metode *black box testing* berbasis *equivalence partitioning*. Metode ini dipilih sesuai dengan tujuan pengujian yaitu untuk memeriksa fungsionalitas dari *website* dengan membandingkannya dengan *user requirement* untuk dapat mempertemukan ekspektasi *user* dengan sistem yang dibangun. *Equivalence partitioning* akan membagi masukkan data ke dalam kelas-kelas partisi yang memiliki perilaku yang sama dan diharapkan dapat menghasilkan keluaran yang sama [4]. Teknik ini dipilih karena dalam proses pengujiannya dapat dilakukan eksplorasi untuk semua kemungkinan dalam fungsi *website* sehingga dapat menemukan kesalahan yang tersembunyi dan mampu untuk mengurangi jumlah *test case* yang akan diuji.

Pengujian *black box testing* akan dijalankan dengan pola eksekusi secara manual dan otomatis dengan *tool* Katalon Studio untuk melihat bagaimana perbedaan penggunaan kedua teknik pengujian dalam implementasinya. Katalon Studio adalah solusi pengujian otomatis yang dikembangkan oleh Katalon LLC yang dirilis secara publik pada September 2016. Katalon studio merupakan sebuah perangkat lunak pengujian yang bersifat *open-source* yang dibangun di atas *framework* Selenium, Appium untuk pengujian Web, Mobile, API atau Web Services dan desktop menggunakan pendekatan *key-driven* dan *data-driven*. Katalon studio dipilih sebagai *tool* pengujian pada penelitian ini karena katalon studio memiliki dokumentasi yang cukup lengkap dan mudah ditemukan, tidak memerlukan instalasi *browser driver* karena telah disediakan oleh katalon dengan berbagai pilihan *browser*, tampilan UI yang *user friendly*, pengembangan *test script* yang beragam yaitu melalui mode *recording*, penulisan *script* dan manual dengan input *keyword* serta katalon dapat diintegrasikan dengan github yang memudahkan kolaborasi di dalam tim [5].

## 2. Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan pengumpulan informasi dari berbagai sumber seperti buku, jurnal, dan skripsi yang telah disusun oleh peneliti-peneliti terdahulu mengenai pengujian perangkat lunak yang akan digunakan sebagai bahan perbandingan dan referensi dalam proses pengujian yang dilakukan penulis. Pengujian dengan metode *black box testing* adalah metode yang digunakan untuk menguji fungsionalitas dari *website* untuk mengetahui apakah fungsi-fungsi berjalan sesuai dengan yang diharapkan terjadi. Metode ini juga yang diterapkan oleh Rully Pramudita pada tahun 2020 pada penelitiannya yang berjudul “Pengujian *Black Box Testing* pada Aplikasi E-campus Menggunakan Metode *Equivalence Partitioning*”. Penelitian tersebut bertujuan untuk menguji setiap fungsi *website* yang mengelola sistem akademik dan keuangan perguruan tinggi. Dari hasil pengujian, didapatkan hasil bahwa pengujian terhadap aplikasi E-campus masih memiliki kekurangan dalam hal validasi kesalahan yang ditampilkan sehingga perlu segera dilakukan perbaikan [6].

Penelitian lainnya juga dilakukan oleh Yudie Irawan, dkk pada tahun 2019 yang menggunakan metode *black box testing* berbasis *equivalence partitioning* untuk menguji fungsi *website* yaitu penyimpanan data pada manajemen bank sampah. Pengujian dilakukan secara manual dengan pembuatan *test case* dan menentukan partisi data. Dari hasil pengujian, didapatkan hasil bahwa terdapat fungsi yang tidak berjalan dengan *requirement* karena data yang diinput dapat melebihi nilai partisi data yang telah disusun. Dalam penelitian ini juga, peneliti dapat membuktikan bahwa dengan menerapkan teknik *equivalence partitioning* dapat mengurangi jumlah *test case* yang diuji karena hanya perlu melakukan pengujian pada sampel dari setiap partisi yang telah disusun [7].

Penelitian serupa dilakukan juga oleh N. Rahadi, dkk pada tahun 2020 yang berjudul “Pengujian *Software* Aplikasi Perawatan Barang Milik Negara Menggunakan Metode *Black Box Testing Equivalence Partitions*”. Pengujian dilakukan pada fungsi penggantian barang dengan menerapkan empat tahap pada teknik *equivalence partitioning* yaitu penyusunan *test case*, membuat partisi, menyusun data uji dan eksekusi. Hasil akhir terhadap pengujian aplikasi tersebut menunjukkan bahwa tingkat keefektifan dari fungsional aplikasi dalam modul penggantian barang adalah sebesar 93.2% [8].

Pada tahun 2019. A. Arwaz, dkk melakukan penelitian dengan judul “Pengujian *Black Box* pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik *Equivalence Partitioning*”. Pada penelitiannya, digunakan masukkan data acak yang bertujuan untuk memastikan apakah sistem menolak untuk menyimpan data masukan pada database yang tidak sesuai dengan *requirement*, sehingga sistem dapat dikatakan layak untuk digunakan. Hasil akhir dari pengujian ditemukan masih banyak kesalahan pada validasi data yang dimasukkan dan ini dapat membuat data tidak akan tersimpan dengan format yang benar di dalam database [9].

Penelitian lainnya yang berjudul “*Automation of Android Application Testing PT. Bareksa Portal Investasi Using Appium Framework*” yang dilakukan oleh Johan untuk menemukan aplikasi yang sesuai untuk pengujian aplikasi pembiayaan milik PT. Bareksa Portal Investasi karena pengujian yang berulang pada aplikasi yang kompleks tidak mungkin dilakukan secara manual dan untuk menghindari terjadinya *human error*. Pengujian yang dilakukan pada beberapa fungsi dalam aplikasi android tersebut ternyata menjawab masalah dalam hal fitur tambah dan algoritma uji yang dapat disesuaikan langsung dengan bahasa pemrograman yang digunakan dalam *Appium Framework* [10].

### 3. Metodologi Penelitian

Penyusunan penelitian ini memiliki enam tahapan yaitu: (1) Tahap studi pustaka yang bertujuan untuk mendapatkan gambaran mengenai dasar-dasar pengujian dengan menggunakan metode *black box testing* berbasis teknik *equivalence partitioning*, sehingga dapat mengambil langkah yang tepat dalam penelitian ini. Metode ini dipilih berdasarkan batasan pengujian yaitu pada fungsionalitas *website*. (2) Tahap analisis alur fungsi dalam *website* yang bertujuan untuk dapat memahami alur fungsi dari *website Acc.co.id Revamp* yang akan diuji dengan mempelajari dokumen yang ada yaitu *Functional Design Revamp* dan *Draft User Story* dan untuk memberikan strategi yang tepat dalam pengujian untuk masing-masing fungsi dalam *website*. (3) Tahap merancang *test case* pengujian yaitu tahap untuk membentuk suatu rancangan *test case* yang akan diimplementasikan pada pengujian. Pembentukan *test case* dilakukan agar pengujian yang dilakukan lebih terarah dan dapat mendapatkan hasil yang efektif serta dapat mencakup semua *requirement* untuk diuji. Rancangan *test case* dibentuk dengan melakukan pembedaan terhadap input yaitu *valid* dan *invalid* pada setiap fungsi untuk dapat melihat reaksi keluaran fungsi terhadap input dari pengguna *website*. (4) Tahap melakukan pengujian *website* secara manual yaitu menerapkan pengujian tanpa bantuan *tools* dengan menjalankan rancangan *test case* yang diberi nilai aktual yang bernilai *valid* dan *invalid* dalam proses eksekusinya dan akan dilakukan pencatatan penemuan *bug* untuk dilakukan proses perbandingan hasil keluaran dengan hasil yang diharapkan pada saat fungsi dijalankan. (5) Tahap melakukan pengujian *website* secara terotomatisasi yaitu pengujian secara otomatis dilakukan dengan memanfaatkan *tool* pengujian yang sudah ditetapkan yaitu Katalon Studio. Pengujian dengan katalon dimulai dengan membentuk suatu *script* pengujian dengan menggunakan metode manual yaitu *spy* dan input *keyword* pada *workspace* katalon. Eksekusi *test case* akan dikonfigurasi dengan menggunakan data eksternal dalam proses *binding* dari *file* dokumen berformat excel sebagai nilai masukannya. (6) Tahap penulisan laporan akhir yang merupakan langkah terakhir dari penelitian ini adalah melakukan penulisan laporan akhir yang berisikan hasil dan kesimpulan dari hasil pengujian yang dilakukan dengan metode *black box testing* berbasis *equivalence partitioning* secara manual dan otomatis terhadap *requirement* yang telah ditetapkan.

### 4. Hasil dan Diskusi

Pengujian diawali dengan melakukan perancangan *use case* untuk mengetahui gambaran mengenai kebutuhan fungsional *website* yaitu dengan menyusun *basic* dan *alternative flow*, identifikasi variabel, menyusun matriks skenario dan menyusun matriks *test case*. Matriks *test case* yang telah disusun akan menjadi acuan dalam pemberian nilai aktual pada pengujian manual dan sebagai acuan dalam membentuk data *binding* pada pengujian secara otomatis sehingga dengan adanya rancangan *test case* diharapkan pengujian dapat dilakukan dengan lebih terarah dan dapat mencakup keseluruhan proses pengujian.

Pengujian yang dilakukan pada 15 fungsi yang terdapat dalam *website* Acc.co.id Revamp ditemukan adanya dua fungsi yang tidak berjalan sesuai dengan *requirement* karena kesalahan pada proses validasi masukkan data yaitu pada fungsi tambah simulasi kredit multiguna dan tambah pengajuan produk mobil baru syariah. Kesalahan yang ditemukan menandai bahwa *website* masih memerlukan perbaikan guna keamanan dalam proses pengajuan. Selain itu, dalam penelitian ini penulis melakukan perbandingan waktu pengujian yang dapat dilihat pada Tabel 1.

**Tabel 1. Tabel Persentase Perbandingan Waktu Manual dan Otomatis**

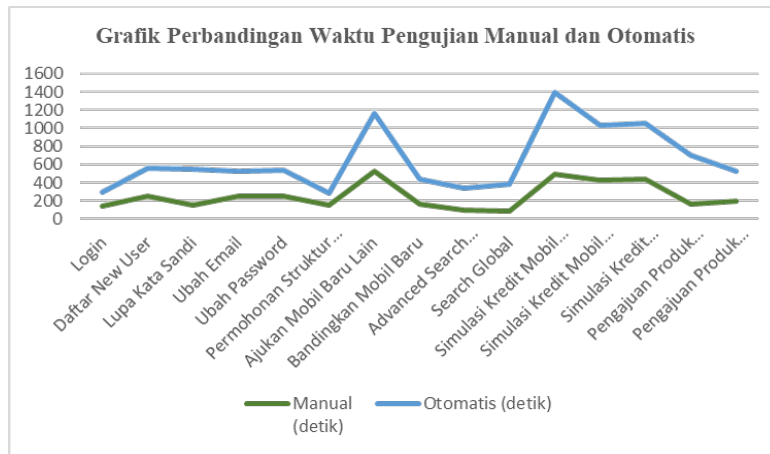
No	Nama Fungsi	Manual	Otomatis	Selisih Waktu	Persentase
1	Login	2 Menit 18 Detik	7 Menit 20 Detik	5 Menit 2 Detik	69%
2	Daftar New User	4 Menit 13 Detik	9 Menit 15 Detik	5 Menit 2 Detik	54%
3	Lupa Kata Sandi	2 Menit 32 Detik	9 Menit 14 Detik	6 Menit 42 Detik	73%
4	Ubah Email	4 Menit 15 Detik	8 Menit 49 Detik	4 Menit 34 Detik	52%
5	Ubah Password	4 Menit 11 Detik	8 Menit 58 Detik	4 Menit 47 Detik	53%
6	Permohonan Struktur Organisasi	2 Menit 35 Detik	4 Menit 42 Detik	2 Menit 7 Detik	45%
7	Ajukan Mobil Baru Lain	8 Menit 44 Detik	19 Menit 24 Detik	10 Menit 40 Detik	55%
8	Bandingkan Mobil Baru	2 Menit 42 Detik	7 Menit 21 Detik	4 Menit 39 Detik	63%
9	Advanced Search Mobil Baru	1 Menit 42 Detik	5 Menit 44 Detik	4 Menit 2 Detik	70%
10	Search Global	1 Menit 28 Detik	6 Menit 28 Detik	5 Menit	77%
11	Simulasi Kredit Mobil Baru	8 Menit 9 Detik	23 Menit 19 Detik	15 Menit 10 Detik	65%
12	Simulasi Kredit Mobil Bekas	7 Menit 12 Detik	17 Menit 17 Detik	10 Menit 5 Detik	58%
13	Simulasi Kredit Multiguna	7 Menit 21 Detik	17 Menit 33 Detik	10 Menit 12 Detik	58%
14	Pengajuan Produk Mobil Baru Syariah	2 Menit 41 Detik	11 Menit 41 Detik	9 Menit	77%
15	Pengajuan Produk Mobil Bekas Syariah	3 Menit 14 Detik	8 Menit 47 Detik	5 Menit 33 Detik	63%
<b>Jumlah Waktu</b>		1 Jam 3 Menit 17 Detik	2 Jam 45 Menit 52 Detik	1 Jam 42 Menit 35 Detik	<b>62%</b>
<b>Rata-rata Waktu</b>		4 Menit 13 Detik	11 Menit 3 Detik	6 Menit 50 Detik	<b>62%</b>

Melalui pencatatan waktu yang telah dipaparkan, dilakukan perhitungan oleh peneliti dengan menggunakan rumus sebagai berikut.

$$\bar{X}_{\text{waktu manual}} = \frac{\text{Jumlah waktu manual}}{\text{Jumlah fungsi}} \quad (1)$$

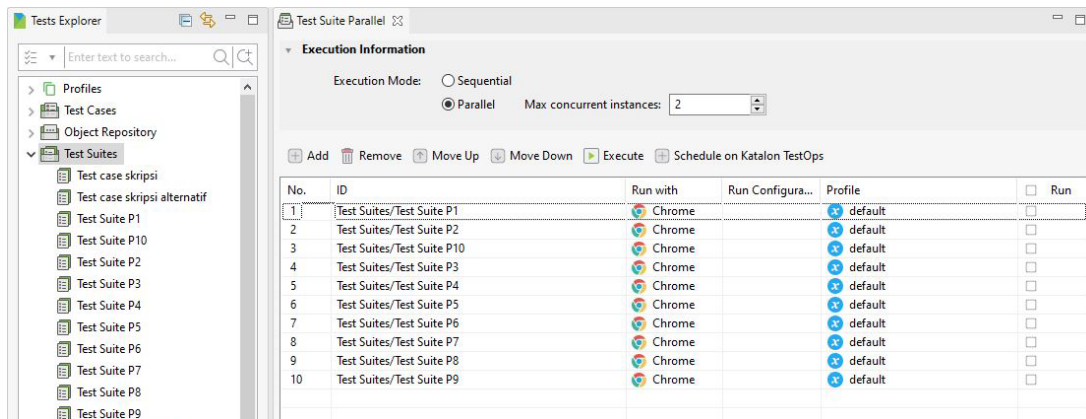
$$\bar{X}_{\text{waktu otomatis}} = \frac{\text{Jumlah waktu otomatis}}{\text{Jumlah fungsi}} \quad (2)$$

$$\text{Persentase waktu pengujian} = \frac{\bar{X}_{\text{waktu otomatis}} - \bar{X}_{\text{waktu manual}}}{\bar{X}_{\text{waktu otomatis}}} \times 100\% \tag{3}$$



Gambar 1. Grafik Waktu Pengujian Manual dan Otomatis

Dari proses perhitungan dengan menggunakan rumus yang dijabarkan, diperoleh bahwa rata-rata lama pengujian manual terhadap pengujian otomatis adalah sebesar 62%. Namun untuk menekan besaran persentase perbedaan lama waktu pengujian manual dan otomatis, penguji melakukan percobaan pada pengujian otomatis dengan menjalankan fungsi-fungsi yang ada ke dalam bentuk pengujian paralel. Pengujian tersebut dijalankan melalui fitur yang telah disediakan Katalon yaitu *test suite collection* yang dapat menjalankan fungsi secara paralel atau menjalankan beberapa fungsi secara sekaligus seperti yang ada pada Gambar 2.



Gambar 2. Test Suite Collection

Fungsi yang dipilih untuk dijalankan secara paralel adalah sebanyak 10 fungsi dengan maksimal fungsi yang dijalankan dalam sekali eksekusi adalah sebanyak dua fungsi. Hal ini bertujuan untuk melihat bagaimana pengaruh pengujian paralel terhadap lama waktu pengujian secara otomatis pada batas iterasi minimal. Dari hasil eksekusi yang dijalankan maka didapatkan hasil seperti yang tertera pada Tabel 2.

Tabel 2. Hasil Pengujian Secara Paralel

No	Fungsi Paralel	Lama Waktu Pengujian	Selisih Waktu
P1	Advanced Search dan Search Global	6 Menit 7 Detik	6 Menit 5 Detik
P2	Bandingkan Mobil Baru dan Permohonan Struktur Organisasi	5 Menit 58 Detik	5 Menit 27 Detik

No	Fungsi Pararel	Lama Waktu Pengujian	Selisih Waktu
P3	Login dan Ajukan Mobil Baru Lain	20 Menit 35 Detik	6 Menit 9 Detik
P4	Ubah <i>Email</i> dan Ubah <i>Password</i>	13 Menit 33 Detik	4 Menit 14 Detik
P5	Daftar dan Lupa Kata Sandi	15 Menit 49 Detik	2 Menit 49 Detik
<b>Total Waktu Penghematan</b>			<b>24 Menit 44 Detik</b>

Total selisih waktu yang didapatkan dari hasil pengujian secara pararel adalah sebesar 24 menit 44 detik lebih cepat jika dibandingkan dengan eksekusi pengujian tanpa mode pararel. Oleh karena itu, melalui hasil yang didapatkan tersebut penguji dapat menekan angka persentase perbedaan pengujian manual dan otomatis hingga 7% seperti yang tertera pada Tabel 3 dan mode ini dapat dimanfaatkan untuk dapat memaksimalkan waktu pengujian secara otomatis.

**Tabel 3. Hasil Perhitungan Persentase Pengujian Manual dan Pengujian Pararel**

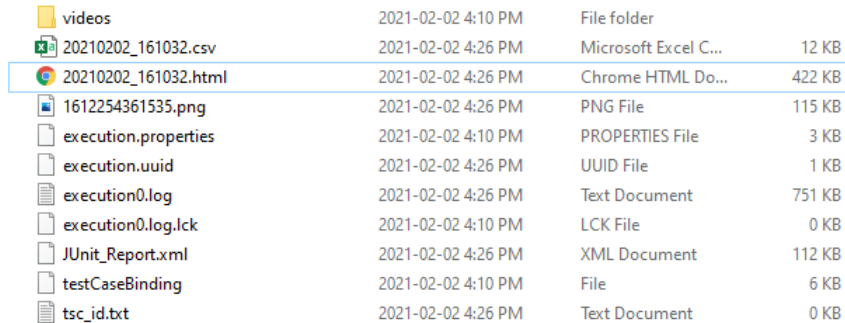
	Manual	Otomatis	Selisih Waktu	Persentase
<b>Jumlah Waktu</b>	1 Jam 3 Menit 17 Detik	2 Jam 20 Menit 39 Detik	1 Jam 17 Menit 22 Detik	<b>55%</b>
<b>Rata-rata Waktu</b>	4 Menit 13 Detik	9 Menit 22 Detik	5 Menit 9 Detik	<b>55%</b>

Besaran persentase yang menunjukkan waktu kecepatan pengujian manual lebih baik jika dibandingkan dengan pengujian otomatis tersebut berlaku jika jumlah *test case* yang diuji tidak memiliki jumlah yang cukup banyak dan pada penelitian ini penguji menggunakan jumlah *test case* dengan rata-rata sebanyak 15 data, sehingga dapat dikatakan waktu proses pengujian tidak dapat menjadi patokan bahwa pengujian manual selalu lebih baik dari pengujian otomatis dan hal itu tergantung dengan jumlah *test* data yang diuji dalam setiap proses pengujian yang dilakukan. Proses yang lama dalam proses pengujian dengan menggunakan katalon studio dikarenakan terdapat pemakaian *keyword* objek yang memiliki fungsi yang menjalankan waktu tunggu untuk melakukan *load website* dan verifikasi pada *test object* seperti *wait page for load*, *delay*, dan *verify element*.

Analisis lebih lanjut dilakukan terhadap perbedaan penerapan pengujian secara manual dan otomatis. Dalam penerapan masing-masing pengujian, penulis mengamati beberapa kendala dan keuntungan yang terdapat pada kedua teknik pengujian. Kendala pada pengujian manual terletak pada saat melakukan eksekusi karena diperlukan ketelitian untuk melakukan pengekseseksiannya dan dalam pengujian ini penulis mengalami kendala karena adanya *requirement* dalam *test case* yang belum diuji sehingga *test case* pada fungsi yang terlewat perlu dilakukan pengujian ulang agar dapat menguji keseluruhan *requirement* yang telah ditetapkan *user*. Pengujian manual juga memiliki keuntungan tersendiri dalam penerapannya yaitu dapat memberikan umpan balik mengenai tampilan *website* saat melakukan pengujian fungsional dan dapat melakukan pengujian lebih cepat untuk *test case* yang bersifat sederhana serta dapat melakukan pengujian eksploratif untuk fungsi yang tidak dapat dijangkau oleh pengujian otomatis.

Untuk pengujian otomatis, kendala yang dialami terletak pada saat penyusunan *script* pengujian yaitu sering terjadinya perubahan pada *website* yang mengakibatkan berubahnya *path* pada objek yang telah disusun sehingga perlu dilakukannya langkah *record* atau *spy* ulang agar dapat menjalankan *script* pengujian. Selain itu, dalam proses pengujian diperlukan impor pada kelas tambahan untuk dapat mengidentifikasi objek yang susah dijangkau dengan *keyword* yang disediakan katalon seperti pemanfaatan kelas robot *framework* untuk dapat memberikan kontrol pada *mouse* dan *keyboard*. Keuntungan pengujian otomatis terletak pada Implementasi pengujian yang tidak perlu adanya campur tangan seorang tester sehingga dapat meringankan pekerjaan pengujian. Pengujian otomatis dapat dimanfaatkan pada pengujian berulang dengan *script* yang telah dibangun sebelumnya sehingga dapat menghindarkan dari pengujian yang redundan dan dapat meminimalkan adanya *human error*. Pengujian otomatis juga disertai

dengan laporan yang dapat dihasilkan secara otomatis seperti video, gambar, *file* teks atau online yang dapat dimanfaatkan dalam evaluasi pengujian dan sebagai *tracking bug* yang dapat dijadikan pola pengalaman untuk menemukan kesalahan pada pengujian selanjutnya. Contoh *reporting* yang dapat dimanfaatkan dalam pengujian otomatis dapat dilihat pada Gambar 2.



videos	2021-02-02 4:10 PM	File folder	
20210202_161032.csv	2021-02-02 4:26 PM	Microsoft Excel C...	12 KB
20210202_161032.html	2021-02-02 4:26 PM	Chrome HTML Do...	422 KB
1612254361535.png	2021-02-02 4:26 PM	PNG File	115 KB
execution.properties	2021-02-02 4:10 PM	PROPERTIES File	3 KB
execution.uuid	2021-02-02 4:26 PM	UUID File	1 KB
execution0.log	2021-02-02 4:26 PM	Text Document	751 KB
execution0.log.lck	2021-02-02 4:10 PM	LCK File	0 KB
JUnit_Report.xml	2021-02-02 4:26 PM	XML Document	112 KB
testCaseBinding	2021-02-02 4:10 PM	File	6 KB
tsc_id.txt	2021-02-02 4:26 PM	Text Document	0 KB

Gambar 3. Contoh Hasil *Reporting* pada Katalon Studio

## 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Kesimpulan yang diperoleh dari pengujian terhadap *website* Acc.co.id Revamp dengan menggunakan metode *black box testing* berbasis *equivalence partitioning* adalah: (1) Pengujian dengan teknik *equivalence partitioning* berhasil dilakukan dengan memberi masukan dengan nilai *valid* dan *invalid* serta menemukan 2 (dua) *bug* dari 15 fungsi yang masih belum memenuhi *user requirement* yaitu *button* yang tidak *disabled* saat *user* melakukan inputan data yang bernilai *invalid* sehingga diperlukan perbaikan guna menjaga keamanan dalam proses pengajuan baik simulasi atau pengajuan produk syariah. (2) Pengujian secara manual dan otomatis memiliki perbedaan dalam menguji *website* Acc.co.id Revamp. Pengujian manual meski memiliki lama waktu pengujian yang relatif cepat tetapi dalam prosesnya rentan untuk terjadi kesalahan dan *test case* yang terlewat yang tidak dilakukan pengujian. Selain itu, untuk pengujian secara otomatis meskipun memiliki waktu pengujian yang lebih lama jika dibandingkan dengan pengujian manual pada pengujian *website* Acc.co.id, pengujian memiliki keuntungan yaitu proses pengujian dapat dilakukan berulang dengan *script* yang sama yang telah dibuat sebelumnya dan jika terdapat perubahan dapat dilakukan sedikit pengaturan untuk mendapatkan hasil yang diharapkan. Pengujian otomatis juga dapat dipilih dalam proses pengujian karena menyediakan laporan yang beragam dan dapat digunakan untuk melakukan evaluasi terhadap fungsi yang telah dijalankan.

### 5.2. Saran

Berdasarkan penelitian yang telah dilakukan, maka pada bagian ini peneliti memberikan saran yaitu: (1) Untuk memperoleh waktu pengujian yang lebih cepat dan efisien pada pengujian otomatis menggunakan katalon studio dapat melakukan percobaan dengan memanfaatkan fungsi *tool* pengujian yang dijalankan secara paralel sehingga hasil yang didapat lebih optimal dan cepat. (2) Untuk meminimalkan kegagalan pengujian yang disebabkan karena *runtime error* seperti kesalahan pada saat memuat halaman *website* pada pengujian otomatis yang dapat menyebabkan timbulnya *failure report*, maka dapat memanfaatkan fungsi *retry* untuk *test case* yang gagal dijalankan agar dapat menguji kembali penyebab kesalahan yang terjadi sehingga hasil yang didapat merupakan murni hasil pada saat fungsi di jalankan. (3) Pada pengujian selanjutnya diharapkan dapat melakukan pengujian dengan memanfaatkan *tool* pengujian yang lebih beragam seperti *tool* pengujian Selenium Webdriver, agar memperoleh hasil pengujian yang lebih akurat dan maksimal

## Referensi

- [1] V. Pujani, "Use of ecommerce websites in developing countries," *World Acad. Sci. Eng. Technol.*, vol. 78, no. April, pp. 790–795, 2011, doi: 10.5281/zenodo.1335265.
- [2] "Digital in Indonesia: All the Statistics You Need in 2021 — DataReportal – Global Digital Insights." <https://datareportal.com/reports/digital-2021-indonesia?rq=digital> 2021 indonesia (accessed Feb. 13, 2021).
- [3] F. Salehi, B. Abdollahbeigi, A. C. Langroudi, and F. Salehi, "The Impact of Website Information Convenience on E-commerce Success of Companies," *Procedia - Soc. Behav. Sci.*, vol. 57, pp. 381–387, Oct. 2012, doi: 10.1016/j.sbspro.2012.09.1201.
- [4] H. I M A N S H I B A B B A R Page and H. Babbar, "SOFTWARE TESTING: TECHNIQUES AND TEST CASES," 2017, Accessed: Sep. 26, 2020. [Online]. Available: [www.ijrcar.com](http://www.ijrcar.com).
- [5] Z. Ereiz, "Automating Web Application Testing Using Katalon Studio," *Zb. Rad. Međunarodne naučne Konf. o Digit. Ekon. DIEC*, vol. 2, no. 2, pp. 87–97, 2019.
- [6] R. Pramudita, "Pengujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 2, p. 100, 2020, doi: 10.32493/jtsi.v3i2.4695.
- [7] Y. Irawan, S. Muzid, N. Susanti, and R. Setiawan, "System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa)," Feb. 2019, doi: 10.4108/eai.24-10-2018.2280526.
- [8] N. W. Rahadi and C. Vikasari, "Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions," *Infotekmesin*, vol. 11, no. 1, pp. 57–61, 2020, doi: 10.35970/infotekmesin.v11i1.124.
- [9] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.
- [10] Johan and R. Gonydjaja, "Automation of Android Application Testing PT. Bareksa Portal Investasi Using Appium Framework," *Int. Res. J. Adv. Eng. Sci.*, vol. 4, no. 2, pp. 252–257, 2019, Accessed: Oct. 03, 2020. [Online]. Available: [www.appium.io](http://www.appium.io).