

Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale

(Studi Kasus TPOS PT. JAVASIGNA INTERMEDIA)

Yoga Kosasih
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
16523081@students.uii.ac.id

Andhik Budi Cahyono
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
andhikbudi@uui.ac.id

Abstract— Tidak seperti *Blackbox Testing* yang biasanya melakukan testing secara manual, dalam penelitian ini menggunakan *automation testing tool* yaitu Katalon. Katalon bisa digunakan untuk pengujian dengan menggunakan *test case*. *Test case* ini dilakukan *record and playback*, yaitu dengan merekam setiap respon dalam satu event pada aplikasi yang digunakan. Tujuan dari penelitian ini adalah untuk menganalisis efektivitas aplikasi yang sudah dibangun, yang kemudian apakah *Automation testing tool* Katalon lebih efektif jika dibandingkan dengan pengujian manual. Penelitian ini menjelaskan mekanisme *automatic waiting* untuk memainkan perintah tes. Dan hasil percobaan menunjukkan bahwa waktu tunggu dapat ditentukan secara otomatis dan dinamis sehingga pengujian tidak harus menambahkan perintah tunggu secara manual sehingga mengurangi waktu dan kesalahan pembuatan.

Keywords—*Katalon Studio, Software testing, Software Assurance, Blackbox testing, Web testing, Point Of Sale*

I. PENDAHULUAN

TPOS merupakan aplikasi website untuk mengelola penjualan produk pada tiap-tiap titik toko di bermacam daerah di Indonesia. Secara teori POS (*Point of sale*) merupakan tempat di mana pelanggan melakukan pembayaran untuk barang atau jasa, dan di mana pajak penjualan dapat dibayarkan. Hal itu bisa dilakukan di tempat toko fisik, di mana terminal POS dan sistem digunakan untuk memproses pembayaran kartu, atau titik penjualan virtual, seperti komputer atau perangkat elektronik seluler [1]. TPOS diharapkan dapat membantu transaksi penjualan dan perhitungan stok barang. Para admin toko dapat melakukan penjualan dan perhitungan stok barang melalui TPOS dari gawai mereka. Manfaat lain yang diharapkan adalah kemudahan admin untuk mengelola toko diberbagai titik daerah di Indonesia melalui satu aplikasi mulai dari informasi penjualan hingga stok produk.

Beberapa makalah telah membahas pengembangan sistem web dan contohnya adalah Pembangunan Sistem Informasi Point Of Sales Terintegrasi Dalam Lingkup Rumah Makan Beserta Cabangnya (Studi Kasus: RM. Pecel Pincuk Bu Tinuk) [2]. Makalah tersebut menjelaskan proses pengembangan sistem POS untuk digitalisasi proses konvensional yang sudah ada. Pada sistem yang dibangun telah melalui pengujian fungsional *Whitebox* dan *Blackbox testing* serta pengujian nonfungsional agar sistem dapat dikatakan sudah berjalan dengan baik tanpa harus terjadi

kesalahan yang akan muncul di kemudian hari sehingga web telah dinyatakan telah memenuhi semua kebutuhan sistem [8].

Tujuan penelitian ini adalah untuk mengukur kualitas perangkat lunak berdasarkan metode *Blackbox testing* pada aplikasi *The Point Of Sale* (TPOS). Masalah yang diangkat dalam penyusunan karya ilmiah ini dibatasi pada pengukuran kualitas perangkat lunak berbasis *web* dengan menggunakan metode *Blackbox testing* yang mengacu pada beberapa batasan aspek pengujian yaitu pengukuran perangkat lunak berdasarkan aspek *functional test*, pengukuran perangkat lunak berdasarkan aspek *Frontend Performance test*, pengukuran perangkat lunak berdasarkan aspek *Backend Performance test*, dan pengukuran perangkat lunak berdasarkan aspek *User Interface test*.

Fokus pembahasan karya ilmiah ini adalah katalon studio pengujian aplikasi web otomatis merupakan hal yang penting topik penelitian dalam rekayasa perangkat lunak. Itu juga dianggap sebagai salah satu jenis pengujian GUI. Masalah utama yang harus diatasi adalah bagaimana caranya mengotomatiskan pembuatan *test case*.

II. KAJIAN PUSTAKA

A. Katalon Studio

Untuk kebutuhan proses pengujian dilakukan dengan menggunakan *tool* yaitu Katalon Studio. *Tool* tersebut merupakan aplikasi open source untuk melakukan pengujian secara otomatis yang telah dikembangkan oleh Katalon LLC yang dapat dijalankan di semua sistem operasi seperti Windows, MAC os dan linux. Katalon Studio menggunakan antarmuka IDE khusus untuk melakukan pengujian [2]. Katalon Studio memiliki tiga fitur utama untuk menunjang beberapa pengujian di beberapa platform seperti Web testing, API testing, dan Mobile testing. Selain itu Katalon studio telah terintegrasi dengan beberapa teknologi dari luar seperti github. Katalon Studio sendiri telah menyediakan UI grafis yang beragam tampilan, menu, pohon tabel dan lain-lain untuk mengelola *test case* objek dan file data. Walaupun masih dalam tahap pengembangan Katalon Studio juga sudah mendukung beberapa environment seperti browser dan sistem operasi komposisinya sangat ideal untuk pengujian yang hanya ingin melakukan drag-drop dan memiliki kemampuan pemrograman terbatas.

Meski masih dalam tahap pengembangan katalon juga telah menunjang sebagian *environment* seperti sistem operasi dan browser. Dalam pembuatan *test case*, katalon mempunyai 3 metode yaitu merekam aktivitas website, membuat manual *test case* dan membentuk *script*, yaitu *test case* dibuat secara manual oleh penguji

B. Point Of Sale

Sistem POS (*Point of Sale*) ialah suatu sistem untuk menyinkronkan serta mengintegrasikan data pesanan, data reservasi, data kartu hadiah, data e-commerce, ataupun data poin loyalitas yang terletak di fitur POS dengan fitur website orang dagang dan menyinkronkan informasi yang terletak di basis informasi website ke perangkat POS terpaut. Sistem POS digunakan untuk mengintegrasikan sistem *merchant e-commerce* dan sistem pemesanan reservasi, dan lainnya yang disediakan pada situs website. Sistem POS mencakup fitur POS, *server* website POS, susunan basis informasi POS, aplikasi situs website POS.

Adapun karya ilmiah yang berjudul *Web Based Point of Sale System* atau disingkat WPOS berbentuk sistem berbasis website yang membolehkan manajemen laporan toko secara jarak jauh, serta membolehkan pelanggan melakukan penjadwalan ataupun penjadwalan ulang waktu pengiriman [3]. Server menyediakan seluruh informasi dan data penting yang dibutuhkan melalui website browser pelanggan. dalam sistem ini, server antar toko bisa berkomunikasi satu sama lain dengan *mainframe* kantor pusat. WPOS bisa diimplementasikan bagaikan rangkaian terintegrasi untuk kerja sama antar posisi toko.

III. METODOLOGI

Penelitian yang dilakukan tujuannya untuk memperoleh pengetahuan mengenai teknik terbaik pelaksanaan testing dan dirancang dalam beberapa kegiatan

Blackbox testing adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai [5] Pada proses *Blackbox Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan [6].

Adapun *Blackbox testing* menurut *softwaretestinghel.com* (2018) dibagi menjadi dua jenis yaitu (1) *functional testing* dimana jenis ini berkaitan dengan persyaratan fungsional atau spesifikasi aplikasi. Beberapa jenis pengujian fungsional utama yaitu (a) *System Testing*, (b) *Integration Testing*, (c) *Smoke Testing* (d) *Regression Testing* (e) *sanity testing* dan (f) *User acceptance Testing*; dan (2) *Non-functional testing* dimana terlepas dari fungsionalitas persyaratan ada beberapa aspek non-fungsional yang perlu diuji untuk meningkatkan kualitas dan kinerja aplikasi yang meliputi: (a) *Performance testing* (b) *Usability testing* (c) *Load testing* (d) *Compability testing* (e) *Scalability testing*; dan (f) *Stress testing*

Pengujian *Software* sangat penting dilakukan dikarenakan setiap orang membuat kesalahan pada saat membuat software. Kesalahan pada masing-masing *Software* akan berbeda pada masing-masing *software* [4] Maka dari itu perlu dilakukan pengujian *Software* untuk melakukan verifikasi dan validasi

bahwa program dibuat sama dengan kebutuhan dari perusahaan.

Pengujian adalah elemen yang paling penting dalam pengembangan perangkat lunak ini juga bisa menjadi aktivitas kompleks untuk menyusun dengan benar, dan dengan cara yang mendukung efisiensi maksimum yaitu dengan *seven principal of software tester* atau biasa disebut 7 prinsip pengujian perangkat lunak, Berikut adalah penjelasan mengenai tujuh prinsip pengujian:

- **Testing Shows Presence Of Defects** atau pengujian menunjukkan cacat dalam melakukan pengujian perangkat lunak proses yang dilakukan sebenarnya adalah untuk mencari cacat pada sebuah perangkat lunak yang dikembangkan, yang dimana tester harus membuat sebuah perangkat lunak itu menjadi gagal. Itu diperlukan supaya proses pengujian perangkat lunak bisa menemukan kegagalan program sebanyak banyaknya. Meskipun dalam beberapa proses pengujian sangat mustahil untuk memastikan bahwa sebuah perangkat lunak 100% bebas dari cacat/bug, bahkan jika tidak ada cacat yang ditemukan itu bukan bukti kebenaran
- **Early Testing** atau pengujian dilakukan lebih awal pengujian yang dimulai sedini mungkin dalam pengembangan perangkat lunak, sehingga cacat bisa ditemukan oleh tester diawal testing yang nantinya tester bisa merencanakan testing yang akan dibuat itu seperti apa dan ketika menemukan sebuah cacat/bug nantinya tester bisa memberi saran kepada developer
- **Exhaustive Testing Is Not Possible** atau mustahil melakukan pengujian secara menyeluruh melakukan pengujian perangkat lunak secara keseluruhan itu sangat tidak mungkin, tidak perlu melakukan testing secara menyeluruh melainkan bisa juga tester set ekspektasi dan berasumsi membuat skenario batas atas dan batas bawah sebuah perangkat lunak dan tidak perlu mengetes semuanya melainkan menggunakan asumsi bahwa tes sudah dilakukan maka itu akan mencakup keseluruhan
- **Testing Is Context Dependent** atau pengujian bergantung pada konteks pengujian bergantung pada konteks yang pada dasarnya berarti bahwa cara menguji sebuah situs akan berbeda dari cara menguji aplikasi komersial setiap *test case* yang dibuat tergantung konteksnya seperti contoh semisal *test case A* tentang login sedangkan *test case B* tentang logout itu hal yang berbeda dan ekspektasinya juga berbeda baru bisa melakukan testing. Sehingga pada saat proses pengujian pada perangkat lunak harus benar-benar sesuai dengan konteks dari perangkat lunak itu sendiri karena setiap perangkat lunak itu memiliki cara pengujian yang berbeda masing-masing tidak bisa dipukul sama rata
- **Defect Clustering** atau pengelompokan cacat/bug pengelompokan cacat sebuah perangkat lunak pada proses pengujian lebih diutamakan melakukan pengujian pada modul atau fungsional program yang kecil, biasanya semakin kecil modulnya maka cacat/bug yang akan ditemukan akan semakin banyak

- **Pesticide Paradox** atau paradox pestisida proses pengujian yang dilakukan secara berulang pada akhirnya tidak akan menemukan cacat pada perangkat lunak. tester tidak bisa begitu saja bergantung pada teknik pengujian yang ada maka harus terus menerus memperbaiki metode yang ada untuk membuat pengujian lebih efektif seperti melakukan kesalahan membiarkan *test case* tidak di update dan menganggap remeh bahwa *test case* tersebut selalu hijau dan tidak perlu melakukan update *test case* itu hal yang salah dan tidak boleh dilakukan
- **Absence Of Error Fallacy** atau tidak ada bug merupakan kesalahan dalam pengujian tidak ditemukan adanya cacat/bug maka hal itu perlu dicurigai, bisa saja perangkat lunak yang 99% bebas dari cacat masih tidak dapat digunakan, hal ini bisa terjadi jika sistem diuji secara menyeluruh untuk persyaratan yang salah maka pengujian perangkat lunak tidak hanya menemukan cacat tetapi juga untuk memeriksa bahwa perangkat lunak memenuhi kebutuhan bisnis, sering menemukan bahwa tesnya berhasil tapi tidak sesuai dengan fungsi jadi ketika error itu tidak ada bukan berarti aplikasi itu dinyatakan bebas lolos dari QA

Pengujian aplikasi TPOS bisa dieksekusi dengan menggunakan metode *Manual Testing* ataupun dengan metode *Automated software testing* dapat menggunakan salah satu metode atau keduanya untuk kebutuhan test secenario yang kompleks. Kapankah waktu yang tepat untuk menggunakan otomatis testing dan kapan waktu yang tepat menggunakan manual testing? Berikut adalah waktu yang tepat kapan penguji harus menggunakan otomatis testing dan kapan tester menggunakan manual testing:

1. Otomatis Testing

- Tes kritikal dari sebuah produk, tester harus membuat scenario testing untuk core functionality, core functionality biasanya banyak berfungsi pada fungsi fungsi lainnya maka dibutuhkan software testing secara repetitive
- Tes yang harus di eksekusi pada setiap build atau rilis, jika aplikasi yang dimiliki memiliki banyak versi dan harus diuji setiap versinya
- Tes yang harus dijalankan pada banyak kombinasi platform dan os version, seperti 1 skenario di eksekusi untuk platform yang berbeda beda seperti android, ios, web, api dan browser yang berbeda seperti chrome atau Mozilla
- Tes yang dieksekusi pada work flow yang sama tapi menggunakan data yang berbeda beda, seperti contoh semisal ada 1 flow yang selalu di ulang ulang tapi inputan datanya berbeda beda
- Tes yang membutuhkan penginputan data secara masiv dan berulang, contohnya seperti pada saat mengisi form terkadang banyak ada yang berulang juga
- Tes yang digunakan pada performance testing, biasanya akan dilakukan pada performance tes

pada sebuah aplikasi kalau itu pakainya manual bagaimana caranya kita mengeksekusinya

- Tes yang dalam eksekusinya membutuhkan waktu yang lama, biasanya dilakukan smoke test atau tes yang berjalan terus terusan itu harus menggunakan automted tes

2. Manual Testing

- Tes yang hanya dieksekusi 1 kali
- User experience, tes yang membutuhkan user untuk merasakan secara langsung atau mendapat eksperience secara langsung terhadap sebuah alikasi atau real user
- Tes yang harus dilakukan secara emergency, kalau ada bug fix yang harus segera di tes dan kalau menggunakan automation script biasanya akan membutuhkan waktu sedikit lebih lama dibanding manual tes
- Tes yang dilakukan secara random / exploratory testing, ada kalanya sebagai software tester harus bermindset testing shows presence of defects jadi harus explore terus mencari corner case yang belum kita tes
- Tes yang membutuhkan konfirmasi visual, contoh semisal yang ingin pixel perfect buttonya ditengah tengah atau kurang marginya ada ditengah kanan sedikit kiri sedikit terlalu kepinggir 1pixel 2pixel atau gradien warnanya

IV. HASIL DAN PEMBAHASAN

A. Kebutuhan sistem

Kebutuhan minimum sistem untuk menjalankan Katalon Studio diperlihatkan pada tabel1

Tabel 1. Kebutuhan Sistem

	Kebutuhan
OS	Windows 7,8, 10, macOS 10.11+, Linux (Ubuntu)
CPU	Minimum 1 GHz 32-bit (x86) atau 64-bit (x64) Processor
Memori	Minimum: 1 GB RAM (32-bit) atau 4 GB RAM (64-bit) Recommend: 4 GB RAM (32-bit) or 8 GB RAM (64-bit)
Hard Drive	Minimum 1 GB ruang tersedia. Dibutuhkan ruang tambahan bergantung pada proyek dan laporan eksekusi.

Spesifikasi browser yang mendukung katalon studio diperlihatkan pada Tabel 2. Device yang digunakan pada penelitian kali ini, hanya berfokus pada desktop

Tabel 2. Spesifikasi Browser

Desktop Browsers	Version on Windows	Version on macOS
Internet Explorer	9, 10, 11	
Microsoft Edge	Current	
Firefox	56+	Untuk Firefox 57, gunakan Katalon Studio v5.1+
Google Chrome	58+	
Opera	Not supported	
Safari	5.1+	9, 10, 11
Opera	Not supported	

Pengujian dilakukan untuk menguji *respon time* pada aplikasi TPOS dengan menggunakan desktop yang memiliki spesifikasi seperti pada Tabel 3

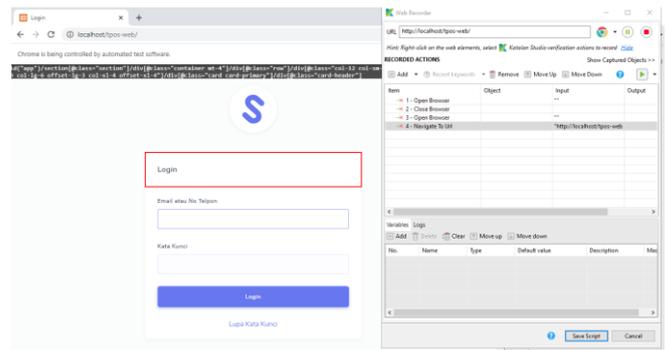
Tabel 3. Spesifikasi Desktop

Komponen	Spesifikasi
Sistem Operasi	Windows 10
CPU	2.50GHz 2.71GHz 64-bit (x64) processor
Processor	Intel® Core™ i5-7200 processor
Memori	8GB RAM
Hard Drive	1 TB
Koneksi Internet	10 Mbps

B. Proses pengujian

Pada penelitian kali ini akan difokuskan kepada *tool* Katalon studio dilakukan dua kali pengujian yaitu pengujian manual dengan menggunakan metode *Blackbox testing* dan pengujian otomatis dengan menggunakan *tool* Katalon studio untuk melihat respon time dari aplikasi yang diuji, hasil *test case* pengujian terbagi menjadi 2 yaitu manual menggunakan Google Spreadsheet dan otomatis menggunakan *test case* yang dibuat dengan metode record & playback yang disediakan oleh katalon. Record/rekam merupakan suatu proses perekaman segala event yang diterima dari seluruh aktivitas pengguna pada halaman web. Seluruh hasil perekaman nanti akan menjadi sebuah test yang akan disimpan oleh katalon, dan nantinya akan dijadikan sebagai *test case*. *Test case* tersebut dapat diganti/modifikasi untuk penyempurnaan proses pengujian, *test case* tersebut

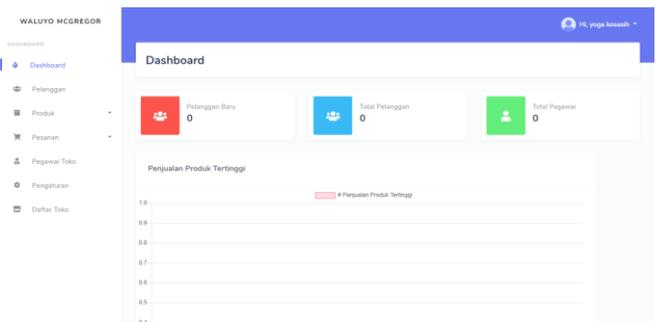
berikutnya dijalankan secara otomatis menggunakan fungsi *playback*. Gambar 1. merupakan proses perekaman pada sebuah halaman *Web TPOS*



Gambar 1. Proses perekaman event pada katalon

Ketika *test case* dijalankan, *log* dari tiap langkah secara otomatis akan terbuat. Log tersebut berisi status status berhasil atau gagalnya suatu langkah dan waktu yang dibutuhkan untuk menyelesaikan langkah tersebut. Ketika terdapat langkah yang gagal maka proses saat menjalankan *test case* akan berhenti dan rincian *test case* tersebut bisa dilihat pada *log view*.

C. Skenario eksperimen



Gambar 2. Dashboard TPOS

Aplikasi TPOS sendiri memiliki berbagai macam komponen seperti pada Gambar 2. bagian bar sebelah samping kiri terdapat dashboard, pelanggan, produk, pesanan, pegawai toko, pengaturan, daftar toko, Dan disamping kanan atas terdapat profil pengguna

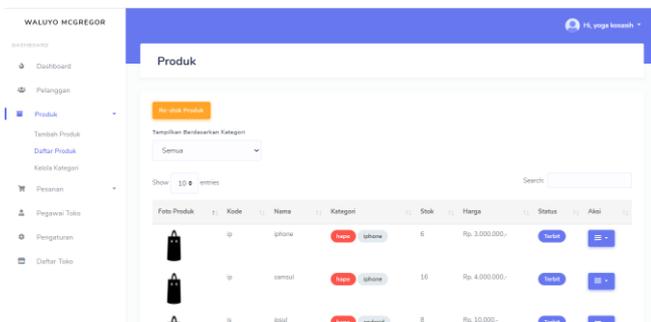
Halaman yang diteliti kali ini adalah halaman untuk menambahkan produk dan pesan produk karena pemesanan produk merupakan fungsi utama dari aplikasi TPOS berikut adalah beberapa skenario pengujian : seperti pada Tabel 4.

Tabel 4. Skenario pengujian TPOS

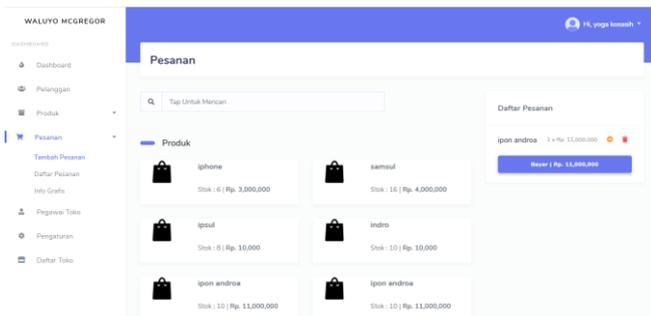
id	Deskripsi Pengujian	Hasil yang Diharapkan
Tc1	Halaman login mengisi Username dengan "yoga.kosasih172@gmail.com" password diisi dengan "bestqa" kemudian klik tombol login atau login menggunakan nomer telpon	Akses login berhasil, sistem akan menutup form login dan menampilkan Dashboard utama TPOS
Tc2	Halaman Tambah produk pengguna mengisi beberapa form seperti kode produk, nama produk, deskripsi produk, kategori produk, bahan,	Tambah produk berhasil dan produk yang telah ditambahkan telah

	ukuran, warna, link terkait, harga, stock produk, dan foto produk	masuk ke daftar Tambah pesanan
Tc3	Halaman Tambah Pesanan pengguna memilih produk lalu membayar produk tersebut	Produk yang dipilih oleh pengguna nantinya akan dibayarkan via toko, Instagram, atau whatsapp
Tc4	Halaman rincian produk pesanan	pengguna akan memilih produk dan lalu melakukan mengatur pemesanan seperti Kurir, No Resi, Tanggal Pengiriman, Alamat Kirim, Metode Pembayaran dan melakukan pembayaran
Tc5	Halaman pembayaran	pesanan yang telah tersimpan tadi akan dibayar Transfer atau Tunai

Gambar 3. Adalah tampilan utama dari halaman Tambah produk aplikasi TPOS Gambar 4. Adalah tampilan utama dari halaman Tambah pesanan aplikasi TPOS.



Gambar 3. Tambah produk



Gambar 4. Tambah Pesanan

Dan masih banyak lagi halaman-halaman yang ditemukan ada kecacatan pada saat pengujian seperti: [Sprint 3] Share produk to WA, [Sprint 3] Remove Pegawai Toko column, [Admin] New feature popups mobile web view di bagian kelola toko + kelola akses, [Admin] Make a Mobile Web View for Admin, [Bug] Add new feature ketika menggunakan web view menu side bar disable, jika menggunakan mobile view menu side bar enable, [Sprint 3] Add Edit Kelola Pengguna, [Sprint 3] Remove unique visitor column, [Sprint 3] Save as Draft on Product, [Sprint 3] Add Nomor Telepon column Kelola Pengguna, [Sprint 3] Add new Infografis Customer Location, [Users] Make a Mobile Web View for Users, [User] Bug aksi di bagian daftar produk, [Users] Bug profil di daftar toko, [User] Bug button di bagian

registrasi, [Invoice] Change date format in Invoice pdf, [Invoice] Add Attachment file when send Email notification, [Users] Bug di bagian login akun, [Users] Add new Menu on Infografis, [Users] Bug nomer telpon di bagian edit pelanggan, [Users] Bug nomer telpon di bagian pegawai toko, [Users] Add Payment Confirmation to Users, [Register] Add +62 when Register using numberphone, [Users] Bug provinsi di bagian buat toko, [Register] Make Register via Numberphone, [Users] Filter Data, [Users] Add new Menu on Pesanan Dibatalkan, [Users] Bug gambar pelanggan di bagian pelanggan, [Users] Add filter to Pegawai, [Users] Update Invoice Header, [Users] Bug icon nomor telpon,email, alamat di edit pegawai, [Users] Bug di foto produk bagian daftar produk, [Users] Bug di tambah sub kategori + edit, [Users] New popups hapus toko di bagian pengaturan, [Users] Bug pindah ke toko lain di bagian daftar produk, [Users] Bug di bagian daftar pesanan ketika melakukan pembayaran, [Users] Add new akses level aksi di bagian pengaturan, [Users] Bug Kelola Kategori Produk, [Users] Bug nama, nomer telpon, provinsi di bagian pelanggan, [Users] Update on Subkategori, [User] Add new feature hapus foto profil di bagian edit profile pemilik, [Users] Hide some settings on Pengaturan, [Users] Add +62 to input No. Telepon on input Pelanggan, [Users] Add Filter to Pelanggan, [Users] Add new page Detail on Pegawai, [Users] Add new field on Pegawai, [Users] Tambahkan Halaman Detail Pelanggan di menu Pelanggan, [Users] Change Color on Menu Cetak Pesanan, [Users] Bug on Edit Pelanggan, [Users] Add entries filter on Pesanan Page, [Admin] Manage Data, Ongkir Manual Can Choose Provinsi & Kabupaten, [Users] Add new Metode Pembayaran on Bayar Page, [Users] New Menu Infografis, [Users] Add new register via email/no telpon, [Users] Share Produk to WA, [Users] Bug on Pengaturan, [Users] New view after Timeout Access, [User] Add new notifikasi setelah pesanan di tambah otomatis, [User] Add feature popups di bagian daftar produk, [Users] New popups di bagian daftar toko, [User] Add new feature edit foto profil di bagian edit pegawai + hapus foto profil di bagian edit pegawai, [User] Add new feature hapus foto profil di bagian edit pelanggan, [Users] Bug di tambah pesanan di bagian stok daftar pesanan, [Fix] Remove Columns where export in PDF Excel, [Sprint 3] Change Color to Icon, [User] must number di bagian registrasi menggunakan nomer telpon, [User] new popups di bagian registrasi menggunakan nomer telpon.

Kompleksitas halaman website bisa diukur dari berbagai macam aspek seperti browser, banyaknya class, kecepatan data dan lain-lain. TPOS sendiri pasti memiliki kompleksitas yang sangat tinggi. Dalam penelitian kali ini kompleksitas dapat diukur dengan cara mengamati secara langsung objek yang dimuat dalam suatu halaman

D. Hasil

Pengujian dilakukan dalam beberapa tahapan yang pertama yaitu membuat *test case*. *Test case* yang dibuat digunakan untuk menguji respon time dari halaman utama TPOS, halaman login, dashboard, pelanggan, produk, pesanan, pegawai toko, pengaturan, dan daftar toko. *Test case* untuk pengujian memanfaatkan *test case* yang dihasilkan dari perekaman tiap event yang dicoba penguji pada halaman

tersebut. *Test case* yang telah dihasilkan lalu dimodifikasi kembali karena terdapat event yang tidak bisa terekam oleh katalon seperti *scroll mouse*.

Tahapan berikutnya adalah *test case* tadi yang telah dimodifikasi selanjutnya dieksekusi untuk mendapatkan respon time dari tiap-tiap halaman yang diuji. Eksekusi *test case* di jalankan secara otomatis oleh katalon, *test case* yang digunakan untuk menguji mulai dari halaman utama sampai dengan pesanan produk seperti gambar 5.

Item	Object	Input	Output
➔ 1 - Open Browser		***	
➔ 2 - Navigate To Url		"http://localhost/tpos-web	
➔ 3 - Double Click	input_Email atau No Telpor		
➔ 4 - Set Text	input_Email atau No Telpor	"yoga.kosasih172@gmail.co	
➔ 5 - Set Encrypted Text	input_Kata Kunci_password	"60MO4RNVXgg8T2coUCJ4	
➔ 6 - Send Keys	input_Kata Kunci_password	Keys.chord(Keys.ENTER)	
➔ 7 - Click	input_Email atau No Telpor		
➔ 8 - Double Click	input_Email atau No Telpor		
➔ 9 - Set Text	input_Email atau No Telpor	"yoga.kosasih172@gmail.co	
➔ 10 - Set Encrypted Text	input_Kata Kunci_password	"o4eleAFY7XY=-"	
➔ 11 - Send Keys	input_Kata Kunci_password	Keys.chord(Keys.ENTER)	
➔ 12 - Click	a_Login		
➔ 13 - Click	span_Produk		

Gambar 5. Test case pengujian TPOS

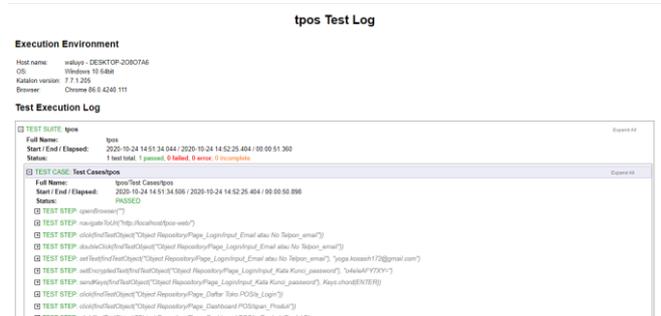
Pada *test case* tersebut item 1 sampai 42 berfungsi untuk menguji respon time mulai dari login sampai pemesanan produk. Tabel 5. Adalah tabel hasil pengujian apakah sudah sesuai dengan yang diharapkan atau tidak sesuai

Tabel 5. Skenario pengujian TPOS

Id	Hasil pengujian	Kesimpulan
T1	Open Browser	Sesuai 15:10:21.640 INFO Starting'Chrome' driver 15:10:21.711 INFO Action delay is set to 0 milliseconds 15:10:26.886 PASSED Browser is opened with url: "
T2	Sistem akan muncul alret berhasil login dan setelah itu masuk ke halaman Dashboard TPOS	Sesuai 15:10:29.146 PASSED Text 'yoga.kosasih172@gmail.com' is set on object 'Object Repository/Page_Login/input_Email atau No Telpor_email'
T3	Produk yang ditambahkan telah berada di halaman Daftar produk	Sesuai 15:10:52.600 PASSED Object: 'Object Repository/Page_Tambah Produk POS/input_Please fill in the first name_submit' is clicked on
T4	Tambah pesanan memilih produk yang akan dibayar melalui toko, Instagram, atau whatsapp	Sesuai 15:10:54.861 PASSED Object: 'Object Repository/Page_Produk POS/a_Tambah Pesanan' is clicked on
T5	Pesanan yang telah tersimpan di Daftar pesanan lalu memilih produk dan mengatur pemesanan seperti Kurir, No Resi, Tanggal Pengiriman, Alamat Kirim, Metode Pembayaran dan melakukan pembayaran	Sesuai 15:10:57.941 PASSED Object: 'Object Repository/Page_Order POS/td_w-300920 153845' is clicked on
T6	Melakukan pembayaran Transfer atau Tunai	Sesuai

		15:10:57.941 PASSED Object: 'Object Repository/Page_Order POS/td_w-300920 153845' is clicked on
T7	Close Browser	Sesuai 15:11:01.550 PASSED Browser is closed

Total keseluruhan fungsi yang diuji ada 5 fungsi dan 42 *item/object* yaitu mulai dari pada saat pengguna login, tambah produk, tambah pesanan, pesanan yang tersimpan, lalu melakukan pembayaran hingga pengguna berhenti melakukan testing, jumlah fungsi yang ditemukan *error* 0 atau tidak ditemukan adanya *error* pada saat pengujian, Gambar 7. Menunjukkan response time yang diperoleh dari proses pengujian menggunakan *tool* Katalon dan bisa dikatan *test case* tersebut berhasil karena *test case* semuanya berwarna hijau, Gambar 8. *Test case* manual diperoleh dari proses pengujian manual bisa dibilang bahwa *test case* tersebut juga berhasil



Gambar 7. Automation Test case

Gambar 8. Manual Test case

E. Pembahasan

Dari segi kompleksitas hampir semua memiliki kompleksitas yang seimbang semua memproses suatu proses rata-rata 10 detik dengan total keseluruhan 41.206s. Dalam pengujian dengan menjalankan *test case* respon tiap *test case* berbeda-beda apabila dijalankan beberapa kali.

Katalon memiliki fitur record/merekam agar membantu penggunaanya dalam membuat *test case* agar lebih mudah dan katalon sangat sensitif terhadap error pada *test case* terkadang dalam merekam aktivitas untuk *test case* banyak event yang coba dilakukan oleh user yang tidak berjalan dalam *test case* atau banyak bagian-bagian yang terekam oleh sendirinya. Maka *test case* tersebut perlu diperbaiki secara manual agar *test case* berjalan dengan semestinya.

V. KESIMPULAN

Proyek *The Point OF Sale* atau TPOS studi kasus PT. JAVASIGN bisa dibilang belum selesai sepenuhnya dikerjakan atau masih dalam proses pengembangan kurang lebih *on progres* 70-80%. Berdasarkan eksperimen yang dilakukan dapat diambil kesimpulan bahwa kesimpulan tergantung dari kebutuhan testing yang dilakukan jika membutuhkan pengujian yang repetitif atau harus menguji banyak platform data yang besar sebaiknya itu di buat automation script dan jika sifatnya cuma 1 kali testing atau dibutuhkan perasaan atau eksperience langsung biasanya sifatnya emergency sebaiknya manual juga bisa dipertimbangkan dan sangat disarankan menggunakan ram 8gb dikarenakan berdasarkan pengalaman pada menggunakan ram 4gb saat melakukan *playback* katalon akan *stuck* dan pengujian yang sedang berlangsung selalu menunjukkan hasil fail/gagal, setelah di upgrade ram menjadi 8gb sejauh ini katalon tidak mengalami kendala sama sekali.

Katalon studio sendiri dibangun oleh perusahaan sebagai *bundle* terpadu yang mencakup hampir semua hal yang diperlukan seperti java, android SDK, web driver. Katalon juga menyediakan pembuatan test case dalam mode scripting, template, kerangka kerja, terintegrasi fitur Jira, Git, Jenkins, dll membantu mempercepat waktu pembuatan test case. Pengguna hanya perlu fokus pada otomatisasi skrip pengujian. Skrip yang direkam dalam satu browser dapat dibuat untuk berjalan disalah satu browser yang didukung membuat pengujian lintas-browser cepat dan mudah. Laporan hasil pengujian intuitif secara grafis dan dapat diekspor ke format pdf dan CSV. Hasil percobaan menunjukkan bahwa waktu dapat ditentukan secara dinamis dan otomatis sehingga tester tidak perlu menambahkan secara manual. Mekanisme yang diusulkan diimplementasikan sebagai bagian dari pengujian web open source dan telah diadopsi lebih lanjut oleh katalon studio dan pada saat telah banyak tester di seluruh dunia menggunakan mekanisme ini untuk pengujian aplikasi *web*.

VI. REFERENCES

- [1] C. A. Sukandar, "Warta Ekonomi," 23 April 2019. [Online]. Available: <https://www.wartaekonomi.co.id/read224883/apa-itupoint-of-sale.html>. [Accessed 18 Desember 2019].
- [2] A. S. Sani, F. Pradana and D. S. Rusdianto, "Pembangunan Sistem Informasi Point Of Sales Terintegrasi Dalam Lingkup Rumah Makan Beserta Cabangnya (Studi Kasus: RM. Pecel Pincuk Bu Tinuk)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, pp. 3249-3257, 2018.
- [3] Katalon LLC, "Katalon Documentation", 2018, [Online]. Tersedia : <https://docs.katalon.com/katalonstudio/docs/index.html> [20 juni 2020].
- [4] M. Manno, "Web Based Point of Sale System". United States Patent US 2004/0181454 A1, 16 September 2004.
- [5] Latif, A. (2015). Implementasi Kriptografi Menggunakan Metode Advanced Encryption Standar (AES) Untuk Pengamanan Data Teks. *Jurnal Ilmiah Mustek Anim*, 4(2), 163-172.
- [6] Akiladevi, R., Vidhupriya, P., & Sudha, V. (2018). Software Testing Tools. *International Journal of Pure and Applied Mathematics*, 118(18), 1783-1800.
- [7] Wahyudi, R., Utami, E., & Arief, M. R. (2016). Sistem Pakar E-Tourism Pada Dinas Pariwisata D.I.Y Menggunakan Metode Forward Chaining. *Jurnal Ilmiah DASI*, 17(2), 67-75.

- [8] Softwaretestinghelp.com. 2018. Black Box Testing: An In-depth Tutorial with Examples and Techniques. <https://www.softwaretestinghelp.com/black-box-testing/>, diakses tanggal 25 Agustus 2018.

VII. LAMPIRAN

Reviewer A

1. Belum ada penjelasan alasan perlunya pengujian otomatisasi. Masalahnya apa? tujuannya apa? cakupannya apa?
2. Kajian literatur tidak jelas rujukannya. Tidak ada perbandingan atau ulasan dari penelitian terdahulu
3. Metodologi tidak jelas, hanya berisi uraian teori. Kontekstualisasikan dengan tahapan yang akan anda kerjakan
4. Hasil tidak diuraikan secara runut dan sistematis. Mengapa skenario pengujian hanya pada halaman web tersebut? apa itu sudah mewakili semua halaman di TPOS? kalau tidak mengapa? mengapa hanya respon time saja. Apa tidak ada temuan kesalahan aplikasi TPOS?
5. Kesimpulan kok membahas perbandingan dengan tools lainnya? kesimpulan belum menyinggung tujuan, kesesuaian hasil dengan tujuan dan saran
6. Tidak ada satupun kalimat paragraf yang merujuk bagian referensi. Referensi yang digunakan sangat sedikit dan tidak cukup menjadi landasan penelitian. Padahal kalau mau googling, banyak referensi tersedia.

Saran saya, cari dan baca jurnal: "Automation Testing Tool dalam Pengujian Aplikasi Belajar Tajwid pada Platform Android"

Reviewer B

7. Makalah kurang menjelaskan urgensi penelitian. Masalah yang diangkat apa? Mengapa membutuhkan solusi tersebut? Mengapa harus kakas tersebut yang diujikan?
8. Makalah ini sangat kurang menunjukkan/ menampilkan referensi. Sebagai contoh, pada kajian pustakam tidak ada satupun pekerjaan orang lain yang dijadikan bahan rujukan.
9. Makalah tidak menampilkan gap analisis. Pada makalah ini tidak ada tinjauan Pustaka yang sejalan dengan penelitian yang akan dilakukan, sehingga urgensi dan posisi penelitian menjadi tidak jelas. Tambahkan tinjauan Pustaka dari penelitian yang dilakukan oleh orang lain.
10. Metode penelitian yang dilakukan adalah menggunakan metode waterfall, namun tahapan penelitian tidak menggambarkan tahapan penelitian. Gunakan metodologi yang sesuai dengan tahapan pengujian.
11. Pembahasan yang dilakukan sangat kurang. Pembahasan kurang menjelaskan maksud dan tujuan penelitian. Pembahasan dapat dilakukan dengan membandingkan, merubah beberapa skenario pengujian, serta menampilkan perbedaan yang terjadi dari proses pengujian dalam beberapa skenario.
12. Pada kesimpulan tiba-tiba muncul Selenium, di mana hal ini belum dibahas dalam bab manapun. Perlu diselaraskan antara kesimpulan dengan tujuan penelitian yang dilakukan.
13. Pada Referensi, terdapat 4 referensi namun tidak dikutip pada bagian makalah.

Secara keseluruhan artikel ini memiliki potensi, namun penulisannya dirasa sangat kurang. Perlu

perbaikan besar untuk memperbaiki makalah ini agar sesuai dengan kaidah penulisan karya ilmiah.