

Automasi Pengujian Fungsional Aplikasi MyIndibiz Assistant Menggunakan Katalon Studio

Rana Ramadhani Nur Fitriana
Fakultas Informatika,
Universitas Telkom, Bandung
ranaramadhani@students.telkomuniversity.ac.id

Ir. Sri Widowati, M.T.
Fakultas Informatika,
Universitas Telkom, Bandung
sriwidowati@telkomuniversity.ac.id

Villy Satria Praditha, S.Kom.,
M.Kom
Fakultas Informatika,
Universitas Telkom, Bandung
villysatriapraditha@telkomuniversity.ac.id

Abstrak

Pengujian perangkat lunak merupakan tahap penting dalam pengembangan aplikasi untuk memastikan kualitas dan keandalannya sebelum dirilis kepada pengguna. Functional testing menjadi salah satu jenis pengujian utama untuk memverifikasi apakah fungsi dan fitur aplikasi beroperasi sesuai spesifikasi. Aplikasi MyIndibiz Assistant, yang dirancang untuk mendukung agent dalam menjalankan tugasnya menghadapi tantangan dalam pengujian manual yang dilakukan oleh sumber daya terbatas dalam waktu singkat. Proses ini sering kali mengurangi akurasi dan efektivitas pengujian karena tekanan waktu dan beban kerja tinggi.

Sebagai solusi, penelitian ini mengusulkan pendekatan automation testing menggunakan Katalon Studio untuk meningkatkan efisiensi dan akurasi pengujian fungsional. Katalon Studio dipilih karena keunggulannya dalam mendeteksi kesalahan, mempercepat proses pengujian, dan mengotomatiskan validasi perilaku fungsional. Penting untuk memahami kapan pengujian manual tetap diperlukan dan kapan otomatisasi lebih efektif. Pengujian manual tetap memiliki kelebihan dalam aspek tertentu sehingga penelitian ini tidak hanya akan membahas efektivitas pengujian otomatis menggunakan Katalon, tetapi juga akan membandingkannya dengan pengujian manual untuk mengevaluasi kelebihan dan keterbatasan masing-masing metode.

Dengan menggunakan proses pengujian fundamental dari ISTQB, penelitian ini bertujuan untuk mengukur efektivitas dan efisiensi dari proses pengujian fungsionalitas aplikasi MyIndibiz Assistant secara otomatis menggunakan alat uji katalon studio dibandingkan dengan pengujian manual pada proses test implementasi dan eksekusi. Hasilnya proses pengujian yang lebih sistematis, efisien, dan akurat, sehingga dapat mendukung percepatan siklus pengembangan dan peningkatan kualitas perangkat lunak secara keseluruhan

Kata kunci: functional testing, automation testing, manual testing, proses pengujian fundamental

Abstract

Software testing is a crucial phase in application development to ensure quality and reliability before release to users. Functional testing is one of the primary testing types used to verify whether an application's

functions and features operate according to specifications. The MyIndibiz Assistant application, designed to support agents in performing their tasks, faces challenges in manual testing due to limited resources and tight time constraints. This often reduces the accuracy and effectiveness of testing due to time pressure and high workload.

As a solution, this study proposes an automation testing approach using Katalon Studio to enhance the efficiency and accuracy of functional testing. Katalon Studio is chosen for its advantages in detecting errors, accelerating the testing process, and automating functional behavior validation. However, it is essential to understand when manual testing remains necessary and when automation is more effective. Manual testing still has advantages in certain aspects, so this research not only discusses the effectiveness of automated testing using Katalon but also compares it with manual testing to evaluate the strengths and limitations of each method.

By utilizing the fundamental testing process from ISTQB, this study aims to measure the effectiveness and efficiency of functional testing for the MyIndibiz Assistant application through automated testing using Katalon Studio compared to manual testing in the test implementation and execution process. The expected outcome is a more systematic, efficient, and accurate testing process, supporting faster development cycles and overall software quality improvement.

Keywords: functional testing, automation testing, manual testing, process testing fundamental

1. PENDAHULUAN

Latar Belakang

Pengujian perangkat lunak merupakan tahap krusial dalam pengembangan aplikasi untuk memastikan kualitas dan keandalannya sebelum dirilis kepada pengguna[1]. Salah satu metode pengujian yang paling umum digunakan adalah functional testing, yang bertujuan untuk memverifikasi apakah fungsi dan fitur aplikasi bekerja sesuai dengan spesifikasi yang telah ditetapkan[1][2]. Pengujian ini sangat penting karena membantu mendeteksi kesalahan sejak tahap awal sehingga dapat diperbaiki sebelum aplikasi digunakan secara luas.

Aplikasi MyIndibiz Assistant merupakan aplikasi berbasis mobile yang dirancang untuk mendukung Sales Assistant (SA) dan Agent Representative (AR) dalam menjalankan tugas mereka dengan lebih

efisien[3]. Namun, dalam proses pengembangannya, pengujian aplikasi ini masih dilakukan secara manual. Pengujian manual menghadapi berbagai kendala, terutama dalam aspek efisiensi waktu dan ketelitian. Pengujian dilakukan oleh sumber daya yang terbatas dalam waktu yang singkat, sehingga berpotensi mengurangi akurasi dan efektivitas proses pengujian. Beban kerja yang tinggi serta tekanan waktu menyebabkan risiko kesalahan manusia meningkat, yang dapat berdampak pada kualitas akhir aplikasi.

Sebagai solusi untuk meningkatkan efisiensi dan akurasi pengujian fungsional, penelitian ini mengusulkan pendekatan automation testing menggunakan Katalon Studio. Alat ini dipilih karena kemampuannya dalam mendeteksi kesalahan dengan cepat, mempercepat proses pengujian, serta mengotomatiskan validasi perilaku fungsional aplikasi[4][5][6][7]. Dengan otomatisasi, pengujian dapat dilakukan secara konsisten tanpa adanya faktor kelelahan manusia, sehingga dapat meningkatkan efektivitas pengujian secara keseluruhan[4].

Meskipun automation testing menawarkan berbagai keunggulan, penting untuk memahami kapan pengujian manual tetap diperlukan. Pengujian manual masih memiliki kelebihan dalam aspek tertentu, seperti eksplorasi fitur baru, pengujian berbasis pengalaman pengguna, serta skenario kompleks yang sulit diotomatiskan[4]. Oleh karena itu, penelitian ini tidak hanya membahas efektivitas pengujian otomatis menggunakan Katalon, tetapi juga akan membandingkannya dengan pengujian manual guna mengevaluasi kelebihan dan keterbatasan dari masing-masing metode.

Untuk mengukur efektivitas dan efisiensi pengujian fungsional, penelitian ini akan menggunakan proses pengujian fundamental dari ISTQB, yang berorientasi pada konsep dasar pengujian perangkat lunak[8][9]. Penelitian ini akan difokuskan pada tahapan Test Implementasi dan Eksekusi dalam proses pengujian fundamental ISTQB, guna membandingkan pengujian manual dan otomatis yang dilakukan pada aplikasi MyIndibiz Assistant. Diharapkan bahwa penggunaan automation testing dengan Katalon Studio dapat menghasilkan proses pengujian yang lebih sistematis, efisien, dan akurat. Dengan demikian, hasil penelitian ini dapat memberikan kontribusi dalam percepatan siklus pengembangan aplikasi serta peningkatan kualitas perangkat lunak secara menyeluruh.

Topik dan Batasannya

Topik penelitian ini adalah melakukan test implementasi dan eksekusi yang efektif agar dapat memastikan bahwa hasil pengujian telah sesuai dengan product requirement document serta mengukur efektifitas dari pengujian automasi menggunakan Katalon Studio dan pengujian manual pada tahap Test Implementasi dan Eksekusi pada aplikasi MyIndibiz Assistant dengan Batasan penelitian adalah sebagai berikut.

1. Aplikasi yang diuji adalah aplikasi MyIndibiz Sales Assistant dengan version 2.7.0.
2. Aplikasi diuji pada environment staging.
3. Pengujian secara otomatis menggunakan alat uji Katalon Studio.

Tujuan

Tujuan dari penelitian ini adalah untuk melakukan proses pengujian fungsional pada tahapan test implementasi dan eksekusi untuk memastikan bahwa hasil pengujian telah sesuai dengan product requirement document. Tujuan lainnya adalah untuk mengukur efektivitas dan keefisienan dari proses pengujian fungsionalitas aplikasi MyIndibiz Assistant secara otomatis menggunakan alat uji katalon studio dibandingkan dengan pengujian manual.

Organisasi Tulisan

Penelitian ini tersusun dari beberapa bagian. Pada bagian pertama, menguraikan terkait pendahuluan yang terdiri dari latar belakang, topik serta batasan, dan tujuan penelitian. Pada bagian kedua yaitu studi terkait terdiri dari penjelasan mengenai kajian pustaka yang menjadi batasan pengujian. Selanjutnya pada bagian ketiga, menguraikan proses pengujian dengan menggunakan metode terpilih pada penelitian ini. Kemudian bagian keempat menjelaskan tentang hasil dan pembahasan dari hasil pengujian yang dilaksanakan. Terakhir pada bagian kelima berisi kesimpulan dari hasil pengujian yang dilakukan.

2. KAJIAN TEORI

2.1 Software Testing

Pengujian perangkat lunak adalah kegiatan yang bertujuan untuk mengevaluasi program atau sistem dengan tujuan menemukan kesalahan dan menentukan program yang sesuai dengan hasil yang diharapkan sebelum diserahkan kepada pengguna. Tujuan utama dari pengujian perangkat lunak adalah untuk menemukan kesalahan dalam perangkat lunak sehingga dapat diperbaiki dan memenuhi spesifikasi dan kebutuhan pengguna, sehingga kualitas perangkat lunak dapat diukur [10].

2.2 Blackbox Testing

Blackbox testing merupakan teknik pengujian yang mengamati hasil input dan output dari perangkat lunak tanpa perlu mengetahui struktur kode dari perangkat lunak[11]. Tujuan dari blackbox testing adalah untuk mengetahui apakah perangkat lunak telah berfungsi dengan baik. Salah satu teknik pengujian dengan blackbox testing adalah equivalence partitioning. Teknik EQV ini bekerja berdasarkan masukan data setiap form yang memecah domain masukan ke dalam kelompok valid atau tidak valid[11].

2.3 Equivalence Partitioning Testing

Dalam teknik *Equivalence Partitioning* (EP), cakupan (*coverage*) mengacu pada sejauh mana pengujian mencakup berbagai kelas ekuivalen yang telah ditentukan. EP adalah metode pengujian *black-box* yang membagi domain input perangkat lunak menjadi beberapa partisi atau kelas ekuivalen, di mana setiap kelas mewakili sekumpulan input yang diharapkan menghasilkan perilaku serupa. Tujuan utama dari EP adalah mengurangi jumlah kasus uji yang diperlukan sambil memastikan bahwa semua fungsi utama telah diuji[14].

Aspek Cakupan dalam *Equivalence Partitioning*:

1. Cakupan kelas equivalence yang valid : memastikan bahwa setiap kelas ekuivalen yang valid telah diuji setidaknya dengan satu kasus uji.
2. Cakupan kelas equivalence yang valid : memastikan bahwa setiap kelas ekuivalen yang tidak

valid juga diuji untuk memverifikasi bahwa sistem menangani input yang tidak valid dengan benar.

2.4 Functional Testing

Functional testing merupakan salah satu jenis pengujian perangkat lunak yang bertujuan untuk memastikan bahwa aplikasi atau sistem berfungsi sesuai dengan spesifikasi fungsionalnya. Ini melibatkan pengujian fitur-fitur individu dari perangkat lunak untuk memastikan bahwa setiap fungsi bekerja sebagaimana mestinya[1]. Pengujian functional berfokus pada apa yang sistem atau aplikasi seharusnya lakukan.

2.5 Manual Testing

Manual testing merupakan pengujian dengan mengeksekusi test case dengan menjalankan software sesuai dengan skenario yang tertera pada test case yang selanjutnya membandingkan output hasil pengujian dengan hasil pengujian yang diharapkan. Manual testing merupakan kegiatan yang melelahkan yang mengharuskan pengujian memiliki serangkaian kualitas tertentu seperti sabar, jeli, berpikiran terbuka, inovatif dan terampil. Manual testing bisa menjadi sulit untuk dilakukan pada software aplikasi yang besar atau aplikasi yang memiliki cakupan data yang besar[4][13].

2.6 Automation Testing

Saat ini, penggunaan automation testing pada aplikasi sudah sangat pesat. Telah banyak organisasi yang melakukan transisi dari pengujian manual ke pengujian otomatis untuk mempercepat waktu menemukan bug, memberikan feedback dengan lebih cepat, dan memberikan hasil yang akurat serta memungkinkan untuk melakukan pengujian secara berulang yang sulit dilakukan oleh manusia. Automation testing merupakan metode terbaik saat ini untuk melakukan pengujian area fungsional yang luas pada program serta menghemat waktu dan biaya yang dibutuhkan untuk melakukan pengujian aplikasi [4].

2.7 Katalon Studio

Katalon studio merupakan aplikasi open source untuk melakukan pengujian functional secara otomatis [12]. Katalon studio memungkinkan pengujian untuk membuat, menjalankan, dan mengelola skrip pengujian dengan mudah. Tujuan menggunakan Katalon Studio sebagai alat pengujian otomatisasi adalah untuk memudahkan pengujian dalam melakukan aktivitas pengujian. Pengujian dibantu dalam membuat skrip pengujian untuk pengujian yang dilakukan berulang-ulang pada objek yang sedang diuji. Katalon Studio juga membantu dalam mendokumentasikan hasil tes melalui tes melaporkan dan menyediakan proses peninjauan yang divisualisasikan melalui fitur Katalon Analytics[12].

2.8 MyIndibiz Assistant

MyIndibiz Assistant merupakan aplikasi internal Telkom berbasis mobile. Aplikasi ini digunakan oleh sales assistant dan agent representative untuk membantu menyelesaikan dan memantau tugas tugasnya[3]. Aplikasi ini merupakan aplikasi yang masih dalam tahapan pengembangan sehingga akan ada fitur yang baru disetiap versionnya. Untuk version 2.7.0 aplikasi myindibiz assistant memiliki 16 diantaranya login, profile, beranda (monitoring), notifikasi, video tutorial, tambah kompetitor, riwayat poi baru, riwayat poi validasi, riwayat order, beberapa fitur utama yaitu tugas, tambah POI, validasi POI, order, tambah POI sekaligus validasi, refferal dan katalog produk.

Dari semua fitur yang ada di aplikasi MyIndibiz Assistant, hanya 10 yang akan menjadi batasan pengujian yaitu login, profile, beranda (monitoring), notifikasi, video tutorial, tambah POI, tambah POI sekaligus validasi, riwayat POI baru, riwayat POI validasi, tugas. Fitur yang tidak menjadi cakupan pengujian berupa tambah kompetitor, riwayat order, validasi POI, order karena masih terdapat penyesuaian dalam proses pengembangannya yang memerlukan penyempurnaan lebih lanjut sedangkan untuk fitur refferal dan katalog produk masih belum sepenuhnya dikembangkan.

2.9 Proses Pengujian Fundamental ISTQB Certification

Proses pengujian fundamental merupakan serangkaian langkah-langkah terorganisir dan direncanakan yang dijalankan dalam proses pengujian perangkat lunak. Dalam proses ini, cakupannya bersifat fundamental dan universal dalam konteks pengujian perangkat lunak tidak memiliki batasan dalam proyek atau industri tertentu. sejumlah kegiatan dilakukan dengan tujuan untuk meningkatkan mutu produk [8]. proses pengujian fundamental mengidentifikasi tahap-tahap yang khusus dalam proses pengujian yang bertujuan untuk memastikan pengujian perangkat lunak dilakukan secara menyeluruh dan efisien. Dalam pengujian fundamental terdapat beberapa step dalam proses dan aktivitas yang terdiri dari Test planning, Analysis and design, implementation and execution, evaluating exit criteria and reporting, test closure activities [8].

2.9.1 Test Planning

Test planning merupakan tahap awal dalam proses dan aktivitas pengujian fundamental berdasarkan ISTQB. Pada tahapan ini bertujuan untuk merencanakan seluruh kegiatan pengujian secara terstruktur dan strategis[8]. Tahapan ini tim pengujian bertugas untuk membantu membangun rencana pengujian yang diantaranya adalah menentukan tujuan pengujian, batasan pengujian, dan strategi pengujian yang didalamnya terdapat exit criteria.

2.9.2 Test Analysis And Design

Analisis dan desain pengujian adalah aktivitas dimana tujuan pengujian diubah menjadi kondisi pengujian dan desain pengujian yang nyata. Selama analisis dan desain pengujian aktivitas di dalamnya berupa membangun skenario pengujian dan kasus uji[8].

2.9.3 Implementation And Execution

Implementasi dan eksekusi ini merupakan langkah untuk mempersiapkan pengujian dan menjalankan skenario yang telah dirancang[8]. Aktivitas yang dilakukan berupa mengimplementasikan skrip uji dan memastikan semua alat pengujian sudah dikonfigurasi. Dalam hal ini pengujian akan dilakukan dengan alat uji Katalon Studio. Implementasi yang akan dilakukan berupa mempersiapkan Katalon sebagai alat uji, dan memindahkan kasus uji untuk otomatisasi pengujian kemudian membuat test suite dari kasus uji untuk pengujian yang lebih efisien, melakukan eksekusi untuk kasus uji, mencatat hasil pengujian, dan melaporkan cacat jika ditemukan.

2.9.4 Evaluating Exit Criteria And Reporting

Tahap ini memastikan apakah tujuan pengujian telah tercapai dan membuat laporan akhir. Aktivitas yang dilakukan berupa membandingkan hasil pengujian dengan kriteria keberhasilan yang ditentukan (exit criteria),

kemudian dari hasil pengujian dilakukan evaluasi apakah pengujian dapat dihentikan atau perlu dilanjutkan, lalu menyusun laporan pengujian yang berisi temuan dan hasil analisis[8]. Hasil report dari pengujian automation dan manual dianalisis berdasarkan waktu pengujian dan keakuratan yang didapatkan. Setelah hasil uji perangkat lunak tercatat, dilakukan perhitungan akurasi pengujian yang bertujuan untuk membandingkan tingkat akurasi di antara pengujian manual dan terotomasi. Dalam perhitungan nilai akurasi suatu pengujian, nilai akurasi diperoleh dengan menghitung persentase dari jumlah pengujian yang berhasil sukses dibagi dengan total test case yang diuji. Rumus untuk menghitung nilai akurasi pengujian adalah sebagai berikut :

$$Akurasi = \frac{Jumlah\ pengujian\ sukses}{Total\ kasus\ uji} \times 100\%$$

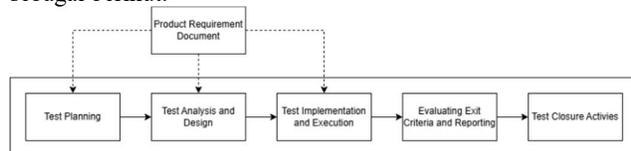
2.9.5 Test Closure Activities

Tahapan terakhir adalah test closure activities yang merupakan aktivitas di mana pelaksanaan pengujian dinilai berdasarkan tujuan yang ditetapkan sebelumnya. Dalam tahapan ini aktivitas yang dilakukan berupa mengumpulkan dan menyimpan hasil pengujian untuk referensi di masa depan, menganalisis pelajaran yang dapat diambil untuk meningkatkan proses pengujian berikutnya, dan memberikan umpan balik kepada tim pengembang atau pemangku kepentingan[8].

3. METODE

3.1 Test Planning

Proses pengujian dengan menggunakan metode proses pengujian fundamental dari ISTQB dengan tahapan sebagai berikut.



GAMBAR 1

Proses Pengujian Fundamental

Tahapan pertama dalam proses pengujian fundamenta adalah Test Planning. Dalam tahapan ini hal yang dilakukan adalah menentukan tujuan pengujian, batasan pengujian yang didalamnya terdapat daftar fitur yang akan di test dan fitur yang tidak masuk dalam cakupan pengujian.

a. Tujuan pengujian

Tujuan pengujian yakni memastikan functionality aplikasi telah berjalan sesuai dengan spesifikasi dan alur skenario pengujian.

b. Batasan pengujian

Berikut merupakan daftar fitur dan deskripsi dari tiap fitur yang masuk dalam cakupan pengujian.

TABEL 1

Daftar fitur cakupan pengujian

No	Fitur	Deskripsi
1	Login	Memungkinkan pengguna untuk masuk ke halaman utama aplikasi My Indibiz SA. Pengguna dengan role SA ataupun AR akan melihat flagging role pada halaman beranda.
2	Beranda	Memungkinkan pengguna untuk melihat semua fitur serta memonitoring jumlah

		validasi POI.
3	Tugas POI	Memungkinkan pengguna untuk melihat daftar tugas POI dan melakukan validasi POI.
4	Tambah POI Baru	Fitur tambah POI baru memungkinkan sales assistant untuk menambahkan POI baru secara langsung pada aplikasi My Indibiz SA. POI baru yang ditambahkan akan masuk ke proses approval untuk selanjutnya di assign menjadi POI yang butuh validasi.
5	Tambah POI Validasi	Fitur tambah POI validasi memungkinkan sales assistant untuk menambahkan POI baru secara langsung pada aplikasi My Indibiz SA sekaligus melakukan validasi POI baru yang ditambahkan.
6	Riwayat POI dibuat	Memungkinkan pengguna untuk melihat riwayat poi baru yang telah dibuat sebelumnya.
7	Riwayat POI validasi	Memungkinkan pengguna untuk melihat riwayat poi yang langsung divalidasi sebelumnya.
8	Video	Memungkinkan pengguna untuk melihat daftar video tutorial penggunaan aplikasi
9	Profile	Memungkinkan pengguna melihat informasi profile
10	Notifikasi	Memungkinkan pengguna melihat daftar notifikasi pengguna

Sedangkan terdapat beberapa fitur dalam aplikasi belum dapat diuji karena alasan tertentu terkait status pengembangannya. yang diantaranya adalah fitur Order, riwayat Order, Tambah Lead, dan Tambah kompetitor tidak akan diuji karena masih terdapat penyesuaian dalam proses pengembangannya yang memerlukan penyempurnaan lebih lanjut. Selain itu, fitur Referral dan Katalog Produk juga tidak termasuk dalam cakupan pengujian, karena fitur-fitur tersebut saat ini belum dikembangkan sepenuhnya. Dengan kondisi ini, pengujian akan difokuskan pada fitur-fitur lain yang sudah stabil dan siap untuk dievaluasi.

c. Exit criteria

Pengujian akan dihentikan atau dianggap selesai apabila seluruh kasus uji yang telah direncanakan telah dijalankan, baik secara manual maupun otomatis dan cakupan pengujian memenuhi setidaknya 95% dari semua persyaratan yang ditentukan. Pengujian akan dihentikan ketika di dalam prosesnya sudah tidak ditemukan defect dengan severity high priority.

3.2 Test Analysis and Design

Tahap ini bertujuan untuk mengidentifikasi kebutuhan pengujian berdasarkan dokumen *Product Requirement Document (PRD)*, menentukan metrik evaluasi, cakupan skenario pengujian, serta menyusun *test case*. Analisis dilakukan dengan merujuk pada PRD untuk menyusun strategi pengujian yang sistematis.

Proses *Test Design* mencakup pembuatan *test case* menggunakan teknik *Equivalence Partitioning*, dengan mencakup *test step*, *test data*, *expected result*, dan *actual result*. Identifikasi data uji juga dilakukan untuk memastikan sistem diuji dengan berbagai variasi skenario yang representatif.

a. Metrik Pengujian

Metrik yang digunakan dalam penelitian ini meliputi *test execution time* dan *coverage pengujian*. *Test execution*

time mengukur waktu eksekusi *test case* baik secara manual maupun otomatis, sementara *coverage pengujian* memastikan semua skenario diuji secara menyeluruh menggunakan teknik *Equivalence Partitioning*. Pengukuran *test execution time* dilakukan dengan mencatat durasi dari eksekusi pertama hingga hasil diperoleh. Pada pengujian manual, waktu dihitung menggunakan *timer*, sedangkan pada pengujian otomatis, waktu diambil dari laporan Katalon Studio. Perbandingan kedua metode ini digunakan untuk mengevaluasi efisiensi otomatisasi dalam proses pengujian. Skenario Testing

Penentuan skenario yang diawali dengan pembuatan user story berdasarkan perilaku pengguna dan fungsionalitas. Penjelasan dari setiap fitur beserta user story dan skenarionya dapat dilihat pada tabel berikut.

TABEL 2
Daftar skenario pengujian

No	Tujuan	Pengujian	Kesimpulan
1	Login	Sebagai SA/AR, saya ingin dapat login dengan mudah ke MyIndibiz Assistant, sehingga saya bisa menggunakan aplikasi MyIndibiz Assistant.	SA dapat login dengan input yang valid
			AR dapat login dengan input yang valid
			SA tidak dapat login dengan input yang tidak valid
			SA dapat melihat pesan error pada kotak kosong
			SA login dengan kata sandi yang tidak valid
			SA login dengan email yang tidak valid
2	Beranda	Sebagai SA/AR, saya ingin masuk ke halaman beranda untuk melihat semua menu dan memonitoring validasi POI, Pengajuan POI dan Pembuatan Order.	SA dapat melihat informasi transformasi digital
			SA dapat melihat halaman beranda
			SA dapat melihat monitoring validasi POI
			SA dapat melihat monitoring pengajuan POI
			SA dapat melihat monitoring pembuatan order
			SA dapat melihat menu coming soon
3	Tugas POI	Sebagai SA/AR, saya ingin dapat melihat semua list tugas dan dapat memvalidasi tugas POI.	SA dapat melihat daftar tugas
			SA dapat memvalidasi penugasan
			SA dapat mengisi identitas POI
			SA dapat mengisi formulir kuesioner
			SA dapat mengisi formulir laporan aktivitas
			SA mengirimkan

			validasi POI
4	Tambah POI Baru	Sebagai SA/AR, saya ingin dapat melihat semua list POI dan menambah POI baru.	SA dapat melihat daftar POI di sekitar pengguna SA dapat menambahkan POI baru
5	Tambah POI Validasi	Sebagai SA/AR, saya ingin dapat menambah sekaligus memvalidasi tugas POI.	SA dapat mengisi formulir tambah POI baru SA dapat melihat halaman konfirmasi SA dapat mengisi formulir kuesioner SA dapat mengisi formulir laporan aktivitas SA dapat menambahkan POI validasi baru
6	Riwayat POI Dibuat	Sebagai SA/AR, saya ingin dapat melihat list dan detail riwayat POI dibuat	SA dapat melihat riwayat POI yang dibuat SA dapat melihat detail riwayat POI yang dibuat
7	Riwayat POI Validasi	Sebagai SA/AR, saya ingin dapat melihat list dan detail riwayat POI validasi	SA dapat melihat riwayat POI yang divalidasi SA dapat melihat detail riwayat POI yang divalidasi
8	Video	Sebagai SA/AR saya ingin dapat melihat list video tutorial penggunaan aplikasi MyIndibiz Assistant	SA dapat melihat daftar video tugas POI SA dapat melihat daftar video buat POI baru SA dapat melihat daftar video tambah order
9	Notifikasi	Sebagai SA/AR saya ingin dapat melihat semua notifikasi berdasarkan filter	SA dapat melihat daftar semua notifikasi SA dapat melihat daftar semua notifikasi berdasarkan tugas POI SA dapat melihat daftar semua notifikasi berdasarkan status POI
10	Profile	Sebagai SA/AR saya ingin dapat melihat informasi profile	SA dapat melihat profil SA dapat melihat informasi profil

Test Cases

Berikut contoh test case berdasarkan fitur login yang menjadi salah satu batasan pengujian.

TABEL 3
Test case fitur login

Test Case ID	Test Scen	Pre- Condi	Test Steps	Test Data	Expe cted	Post Condit	A Pass/ Fail
--------------	-----------	------------	------------	-----------	-----------	-------------	--------------

	ario	tion		Results	ion	-ual Res- ult		
TC_LOG IN_001	SADap at logi n den gan inpu t yan g vali d	Suda h terdaf tar sebag ai agen MyIn dibiz	Given pengg una berada di halam an login And pengg una melih at formul ir login When pengg una mengi si email And pengg una mengi si kata sandi And pengg una mengkl ik tombo l masuk Then pengg una berhas il login And pengg una diarah kan ke berand a My Indibi z Assist ant	<vali d email > <vali d passw ord>	Age n berhas il logi n	Pengg una sekar ang dapat sepen uhnya menga kses MyIn dibiz Assist ant	n/ a	n/a
TC_LOG IN_002	AR dap at logi n den gan	Suda h terdaf tar sebag ai agen	Given pengg una berada di halam an	<vali d email > <vali d passw	Age n berhas il logi n	Pengg una sekar ang dapat sepen uhnya	n/ a	n/a

	inpu t yan g vali d	MyIn dibiz	login And pengg una melih at formul ir login When pengg una mengi si email And pengg una mengi si kata sandi And pengg una mengkl ik tombo l masuk Then pengg una berhas il login And pengg una diarah kan ke berand a My Indibi z Assist ant	ord>		menga kses MyIn dibiz Assist ant		
TC_LOG IN_003	SAD tida k dap at logi n den gan inpu t yan g tidi k vali d	field koson g	Given pengg una berada di halam an login And pengg una mengi si formul ir login When pengg una mengi si email	<inval id email > <inval id passw ord>	Age n meli hat pesa n erro r "Em ail tidi k sesu ai for mat"	Kolo m berub ah menja di merah dan pengg una melih at pesan error "Emai l atau Passw ord yang Anda masuk kan	n/ a	n/a

			And pengg una mengi si kata sandi But email/ kata sandi tidak valid And pengg una mengk lik tombo l masuk Then snack bar muncu l denga n pesan "Emai l/NIK atau Passw ord yang Anda masuk kan salah"		salah"		
TC_LOG IN_004	SAD apat meli hat pesa n erro r pada kota k kos ong	field email koson g	Given pengg una berada di halam an login And pengg una meliha t formul ir login When pengg una mengk lik kotak email And pengg una mengk lik kotak kata	<emp ty email > <emp ty passw ord>	Age n meli hat pesa n erro r "Em ail tida k sesu ai for mat"	Kolo m email berub ah menja di merah dan terdap at pesan error di bawah kolom	n/ a n/a

			sandi And pengg una mengk lik halam an latar belaka ng Then pengg una meliha t pesan kesala han di bawah kotak email And pengg una meliha t pesan kesala han di bawah kotak kata sandi				
TC_LOG IN_005	SAL ogin den gan kata sandi yang tida k vali d	field koson g	Given pengg una berada di halam an login And pengg una meliha t formul ir login When pengg una mengi si email And pengg una mengi si kata sandi But kata sandi tidak valid And	<inval id passw ord>	Age n meli hat pesa n erro r "Em ail tida k sesu ai for mat"	Kolo m berub ah menja di merah dan pengg una melih at pesan error "Emai l atau Passw ord yang Anda masuk kan salah"	n/ a n/a

			pengguna mengklik tombol masuk Then snack bar muncul di bawah kotak email dengan pesan "Email/NIK atau Password yang Anda masukkan salah"					
TC_LOG IN_006	SA login dengan email yang tidak valid	field kosong	Given pengguna berada di halaman login And pengguna melihat formulir login When pengguna mengisi email And pengguna mengisi kata sandi But email tidak valid And pengguna mengklik tombol masuk	<invalid email >	Age n melid	Kolo m email berub ah menjadi merah dan terdapat pesan error di bawah kolom	n/ a	n/ a

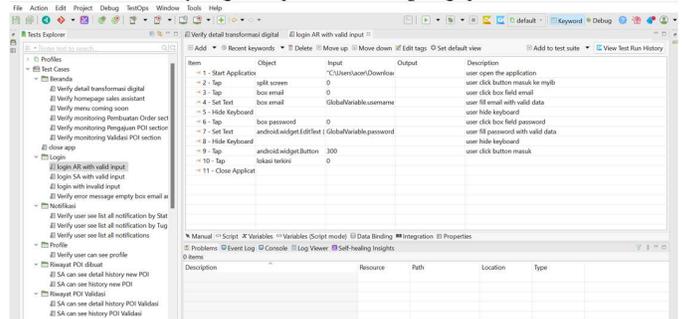
			Then snack bar muncul di bawah kotak email dengan pesan "Email/NIK atau Password yang Anda masukkan salah"					
--	--	--	--	--	--	--	--	--

3.3 Implementation And Execution

Dalam tahapan implementasi dan eksekusi pengujian, test cases yang telah dirancang sebelumnya siap untuk diimplementasikan ke dalam alat uji dan dieksekusi secara manual dan otomatis menggunakan alat uji Katalon Studio.

3.3.1 Implementasi kasus uji pada Katalon

Setelah merancang test case pada tahap sebelumnya, dilakukan implementasi test case ke dalam alat uji Katalon studio. Kemudian mengumpulkan test case pada test suite berdasarkan fitur yang menjadi batasan pengujian.



GAMBAR 2

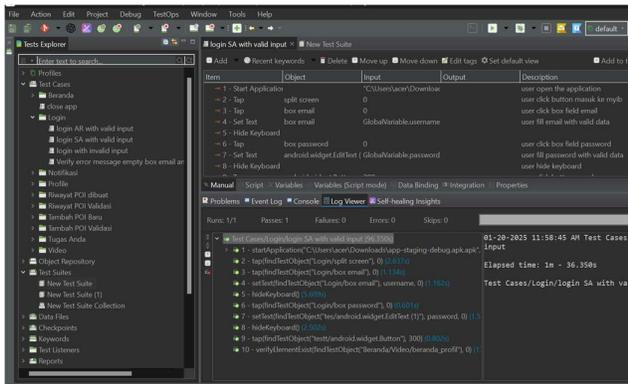
Implementasi test case login

3.3.2 Eksekusi Pengujian Manual

Eksekusi pengujian manual dilakukan dengan membuka aplikasi MyIndibiz Assistant dan melakukan langkah-langkah sesuai test step pada test scenario. Setiap test case yang diuji dihitung waktunya menggunakan timer secara manual lalu dicatat hasilnya. Langkah-langkah dari setiap test case yang dijalankan dihitung waktu mulai dari pertama membuka sampai menutup aplikasinya dan dilakukan sebanyak 5 (lima) kali iterasi.

3.3.3 Eksekusi pengujian

Eksekusi pengujian yang dilakukan pada tiap test case dengan menjalankan skrip pengujian secara otomatis pada katalon studio. Eksekusi pengujian dilakukan pada tiap test suites fitur untuk mendapatkan report pengujian.



GAMBAR 3
Eksekusi test suite

3.4 Evaluating Exit Criteria And Reporting

Setelah proses pengujian berhasil dilaksanakan, tahapan selanjutnya adalah mengevaluasi hasil yang didapatkan dari pengujian secara manual dan automation berdasarkan exit criteria. Hasil pengujian dari manual dan automation testing didapatkan dari pengujian yang dilakukan terhadap 10 fitur yang menjadi batasan pengujian. Evaluasi ini akan membantu dalam mengukur sejauh mana efektivitas masing-masing metode dalam menemukan cacat perangkat lunak, serta dampaknya terhadap efisiensi dan kualitas pengujian. Automation testing akan menghasilkan report dari hasil eksekusi otomatis sedangkan hasil report manual testing dibuat pada dokumen Test Report Summary.

3.5 Test Closure Activities

Tahap ini memastikan seluruh aktivitas pengujian telah sesuai rencana dan hasilnya terdokumentasi untuk referensi mendatang. QA mengevaluasi jumlah *test case* yang dijalankan, defect yang ditemukan, serta status penyelesaiannya. Efektivitas pengujian dianalisis dengan membandingkan hasil aktual dan tujuan awal. Selain itu, *lesson learned* dari pengujian diidentifikasi untuk meningkatkan efisiensi pada proyek berikutnya. Dengan tahap ini, proses pengujian resmi ditutup, dan informasi yang diperoleh dapat digunakan untuk perbaikan di masa depan.

4. HASIL DAN KESIMPULAN

Bagian ini menjelaskan mengenai hasil pengujian yang telah dilakukan berdasarkan skenario uji yang telah dirancang sebelumnya. Pengujian dilakukan dengan tujuan untuk mengevaluasi apakah sistem telah berjalan sesuai dengan spesifikasi yang diharapkan. Setiap hasil yang diperoleh dianalisis berdasarkan teknik pengujian yang digunakan yaitu Equivalence Partitioning.

Pembahasan dalam bab ini mencakup analisis terhadap setiap skenario uji yang telah dijalankan, termasuk status keberhasilan pengujian, temuan yang diperoleh, serta implikasi dari hasil pengujian terhadap kualitas perangkat lunak.

4.1 Hasil Pengujian

Berdasarkan hasil pengujian secara manual dan automation testing terhadap 10 fitur pada aplikasi MyIndibiz Assistant, didapatkan bahwa sebagian besar waktu yang diperlukan untuk melakukan pengujian dengan automation testing lebih cepat daripada manual testing. Dalam segi waktu, pengujian automation lebih unggul dengan total waktu yang dibutuhkan adalah 24.77 menit pada proses

pengujiannya. Sedangkan total waktu yang diperlukan dalam pengujian manual adalah sebanyak 30.42 menit. Berikut tabel hasil analisis perbandingan waktu pengujian manual testing dan automation testing.

TABEL 4

Hasil perbandingan waktu pengujian manual dan automation

No	Fitur	Waktu Pengujian (detik)	
		Manual	Automation
1	Login	319.65	239.946
2	Beranda	217.284	152.598
3	Tugas	306.25	279.742
4	Tambah POI	124.848	115.146
5	Tambah POI Validasi	288.492	267.154
6	Riwayat POI Dibuat	96.672	68.249
7	Riwayat POI Validasi	102.022	80.899
8	Profile	94.244	57.235
9	Video	170.672	105.425
10	Notifikasi	104.93	74.767
Total waktu pengujian (menit)		30.42	24.77

Akurasi kesuksesan pada pengujian manual dan otomatis mendapatkan nilai 100% di beberapa fitur yaitu Login, Beranda, Riwayat POI dibuat, Riwayat Validasi POI, Video, Notifikasi, dan Profile. Sedangkan untuk fitur Tugas POI, Tambah POI baru dan Tambah POI Validasi terdeteksi defect yang mengakibatkan fiturnya tidak dapat dijalankan dengan sempurna dan menjadikan pengujian harus dihentikan berdasarkan exit criteria dengan nilai akurasi kesuksesan pengujian manual dan automation berjumlah 71.67%. Hasil akurasi kesuksesan pada pengujian manual dan otomatis dapat dilihat pada tabel berikut.

TABEL 5

Tabel hasil akurasi pengujian manual dan automation

No	Fitur	Akurasi(%)	
		Manual	Automation
1	Login	100%	100%
2	Beranda	100%	100%
3	Tugas	16.67%	16.67%
4	Tambah POI	0%	0%
5	Tambah POI Validasi	0%	0%
6	Riwayat POI Dibuat	100%	100%
7	Riwayat POI Validasi	100%	100%
8	Profile	100%	100%
9	Video	100%	100%
10	Notifikasi	100%	100%
Total Akurasi(%)		71.67%	71.67%

4.2 Analisis Hasil Pengujian

Melalui tabel hasil perbandingan pengujian manual dan pengujian otomatis didapatkan bahwa waktu pengujian manual lebih lambat dari pada pengujian otomatis. Hal ini membuktikan bahwa pengujian otomatis lebih efisien dalam segi waktu. Efisiensi waktu ini dapat didasarkan oleh pengujian otomatis yang dilakukan dengan alat yang dapat membantu mendeteksi masalah dan bug dengan cepat. Selain itu, pengujian otomatis memberikan konsistensi dan akurasi yang tinggi sebab pada proses yang dijalankan konsisten setiap kali di eksekusi hal ini mampu meminimalisir kesalahan manusia yang mungkin terjadi

saat pengujian manual. Efektif dari pengujian otomatis didapatkan dari kemudahan proses pengujian menemukan semua bug kritis sehingga hasilnya dapat lebih cepat dilaporkan kepada tim sebelum rilis. Hasil laporan dari pengujian otomatis juga lebih sistematis dan dapat langsung digunakan untuk melaporkan temuan bug kepada developer tanpa perlu analisis manual yang memakan waktu. Dengan fitur seperti capture defect dan logging error, QA dapat lebih mudah mengidentifikasi masalah yang terjadi selama proses pengujian.

Perbandingan ini juga menghasilkan perbedaan terhadap proses pengujian secara manual otomasi. Berikut merupakan tabel perbedaan pengujian manual dan automation pada proses pengujian fundamental

TABEL 6

Hasil perbedaan proses pengujian manual dan automation

Proses Testing Fundamental	Pengujian	
	Manual Testing	Automation Testing
Test Planning	Test plan dibuat secara manual, termasuk penjadwalan dan alokasi sumber daya berdasarkan pengalaman tester	Test plan mencakup strategi otomatisasi, pemilihan alat uji (seperti Katalon Studio), dan penjadwalan eksekusi otomatis.
Test Analysis and Design	QA menganalisis dokumen product requirement dan membuat test case secara manual.	Test case dikonversi ke skrip otomatis berdasarkan requirement yang telah dianalisis.
Test Implementation and Execution	QA menjalankan setiap test case secara manual sesuai skenario, membutuhkan interaksi langsung dengan aplikasi.	Skrip otomatis dijalankan oleh alat uji, tanpa campur tangan langsung dari tester, dapat dijalankan secara paralel dan berulang.
Evaluating Exit Criteria and Reporting	Hasil pengujian dicatat secara manual, membutuhkan pencatatan waktu dan identifikasi bug secara subjektif.	Hasil pengujian langsung dicatat dalam laporan otomatis, termasuk waktu eksekusi, log error, dan screenshot bukti uji.
Test Closure Activities	<ul style="list-style-type: none"> • A mendokumentasikan hasil akhir pengujian, menyusun laporan akhir secara manual, serta melakukan evaluasi untuk pembelajaran di masa depan. • E valuasi dilakukan berdasarkan catatan manual, bisa memakan waktu lebih lama dalam merekap temuan dan memberikan laporan akhir. 	<ul style="list-style-type: none"> • A merekapitulasi hasil pengujian dari alat uji otomatis, mengarsipkan skrip dan laporan eksekusi, serta menganalisis efektivitas pengujian otomatis untuk perbaikan selanjutnya. • E valuasi lebih cepat karena laporan otomatis tersedia, memungkinkan analisis efektivitas pengujian secara lebih sistematis.

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa pengujian fungsional secara otomatis menjadi metode yang terbukti lebih efektif dalam mengoptimalkan proses pengujian fungsional, terutama dalam hal konsistensi, kecepatan, dan efisiensi waktu. Dengan otomatisasi, setiap pengujian dapat dijalankan dengan hasil yang lebih akurat tanpa risiko human error, seperti kesalahan input atau kelalaian dalam menjalankan test case. Selain itu, pengujian otomatis memungkinkan eksekusi test case dalam jumlah besar secara paralel dan dapat dilakukan kapan saja tanpa perlu pengawasan langsung, sehingga sangat membantu ketika waktu pengujian terbatas.

Hasil pengujian dengan teknik *Equivalence Partitioning* menunjukkan bahwa *coverage* pengujian dalam otomatisasi lebih baik dibandingkan dengan pengujian manual. Dalam pengujian otomatis menggunakan alat seperti Katalon Studio, setiap partisi yang telah ditentukan dapat diuji secara konsisten tanpa risiko kelalaian, sehingga semua skenario yang mencerminkan partisi yang berbeda dapat dieksekusi secara menyeluruh. Otomatisasi memastikan bahwa setiap eksekusi uji berjalan dengan pola yang sama, sehingga cakupan pengujian lebih terjamin dan dapat diuji secara berulang tanpa perubahan hasil yang disebabkan oleh faktor manusia.

Sebaliknya, dalam pengujian manual, cakupan pengujian tidak selalu terjamin sepenuhnya. Faktor seperti kelelahan, kurangnya ketelitian, atau kelalaian dalam menjalankan skenario uji dapat menyebabkan beberapa partisi tidak diuji sebagaimana mestinya. Selain itu, pengujian manual lebih rentan terhadap variasi subjektif dalam cara pengujian dilakukan, yang dapat memengaruhi hasil uji dan mengurangi efektivitas cakupan pengujian. Oleh karena itu, meskipun teknik EP secara konsep sudah dirancang untuk mencakup semua kemungkinan partisi input, dalam praktiknya, efektivitas cakupan ini lebih optimal dalam pengujian otomatis dibandingkan dengan pengujian manual.

5. KESIMPULAN

Penelitian ini membandingkan proses pengujian pada tahapan Test Implementasi dan Eksekusi untuk meningkatkan efisiensi dan efektivitas pengujian fungsional pada aplikasi MyIndibiz Assistant. Hasil perbandingan menunjukkan bahwa pengujian otomatis menghasilkan pengujian yang lebih cepat serta memiliki *coverage* pengujian yang lebih baik dibandingkan dengan pengujian manual. Beberapa poin utama dari penelitian ini:

- Efisiensi Waktu: Pengujian otomatis mengurangi waktu pengujian secara signifikan dibandingkan manual.
- Akurasi Lebih Baik: Automation testing menghasilkan akurasi lebih tinggi dengan mengurangi kesalahan manusia. Dengan implementasi automation testing, siklus pengembangan aplikasi menjadi lebih cepat dan kualitas perangkat lunak meningkat, mendukung proses validasi fitur dengan lebih efektif.

REFERENSI

[1] M. Niranjanamurthy, S. Navale, S. Jagannatha, and S. Chakraborty, "Functional software testing for web applications in the context of industry," *J. Comput. Theor. Nanosci.*, vol. 15, pp. 1-7, 2018.

- [2] R. Purbaningtyas, "Implementation of functional testing in feasibility testing of the smart malnutrition detection mobile application," *Techno.Com*, vol. 18, no. 3, pp. 251–263, 2019, doi: 10.33633/tc.v18i3.2504.
- [3] *MyIndiBiz Sales Assistant*, Google Play Store, 2025. [Online]. Available: <https://play.google.com/store/apps/details?id=co.id.indibiz.sa>. [Accessed: 01-Feb-2025].
- [4] G. I. Safaat and V. U. Tjhin, "Analysis of quality assurance performance in the application of manual testing and automation testing for software product testing," *Indonesian Interdisciplinary Journal of Sharia Economics (IJSE)*, vol. 7, no. 2, pp. 1987–1996, 2024.
- [5] Bina Nusantara University, "Automating software testing with Katalon Studio: Advantages and limitations," *School of Information Systems*, Nov. 15, 2023. [Online]. Available: <https://sis.binus.ac.id/2023/11/15/automating-software-testing-with-katalon-studio-advantages-and-limitations/>. [Accessed: 01-Feb-2025].
- [6] A. Sivaji et al., "Software Testing Automation: A Comparative Study on Productivity Rate of Open Source Automated Software Testing Tools For Smart Manufacturing," 2020 IEEE Conference on Open Systems (ICOS), Kota Kinabalu, Malaysia, 2020, pp. 7-12, doi: 10.1109/ICOS50156.2020.9293650.
- [7] S. Tjandra, I. Maryati, and J. Theopilus, "Automated software testing for multi platform applications using Katalon Studio," *J. Ilm. Widya Tek.*, vol. 20, no. 1, 2021. ISSN: 1412-7350, eISSN: 2621-3362.
- [8] D. Graham, E. van Veenendaal, I. Evans, and R. Black, *Foundations of Software Testing: ISTQB Certification*, 3rd ed. Boston, MA, USA: Cengage Learning, 2019.
- [9] I. Burnstein, *Practical Software Testing: A Process-Oriented Approach*. New York, NY, USA: Springer Science & Business Media, 2006.
- [10] T. Fulcini, R. Coppola, L. Ardito, and M. Torchiano, "A review on tools, mechanics, benefits, and challenges of gamified software testing," *ACM Comput. Surv.*, vol. 55, no. 14s, Art. 310, Dec. 2023, pp. 1-37, doi: 10.1145/3582273.
- [11] A. Vermal, A. Khatana, and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *NCSE Int. J. Comput. Sci. Eng.*, vol. 5, no. 12, pp. 1-8, 2018.
- [12] Katalon, "Mobile testing made simple," Katalon, 2023. [Online]. Available: <https://katalon.com/mobile-testing>. [Accessed: Feb. 2, 2025].
- [13] R. Haas, D. Elsner, E. Juergens, A. Pretschner, and S. Apel, "How can manual testing processes be optimized? Developer survey, optimization guidelines, and case studies," in *Proc. 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, Aug. 2021, pp. 1281–1291.
- [14] F. D. Hartono and Y. Sugiarti, "Perbandingan metode equivalence partitions dan boundary value analysis pada pengujian black box: Literatur review," *Maj. Ilm. METHODODA*, vol. 12, no. 2, pp. 153–159, 2022. doi: 10.46880/methoda.vol12no2.pp153-159.