

Optimisasi Blackbox Testing Pada Aplikasi Mobile Sicantik Bangsa Dengan Menerapkan STLC (Software Testing Life Cycle)

Sabrina Nur Amira Zain
Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
sabinanuraz@student.telkomuniversity.ac.id

Arief Rais Bahtiar S.Kom., M.Kom.
Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
ariefbahtiar@telkomuniversity.ac.id

Maryona Septiara S.Pd, M.Kom
Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
septiara@telkomuniversity.ac.id

Aplikasi Si Cantik Bangsa merupakan aplikasi mobile pencatatan kesetaraan gender di tingkat desa. Namun, dalam implementasinya masih ditemukan sejumlah bug yang tidak terdeteksi. Penelitian ini bertujuan untuk menemukan bug yang belum terdeteksi melalui penerapan metode black-box testing dengan pendekatan kombinasi manual testing dan automation testing yang terintegrasi dalam kerangka Software Testing Life Cycle (STLC). Pengujian dilakukan terhadap 46 skenario uji yang sama untuk membandingkan efektivitas kedua pendekatan dalam menemukan bug. Hasil pengujian menunjukkan bahwa automation testing mampu mempercepat proses eksekusi, namun manual testing memiliki tingkat deteksi bug yang lebih tinggi. Penerapan STLC dalam proses pengujian berhasil membuat tahapan pengujian lebih terstruktur dan sistematis, sehingga bug yang sebelumnya tidak terdeteksi dapat teridentifikasi. Penelitian ini menunjukkan bahwa pengujian yang terencana dan menyeluruh sangat penting untuk memastikan kualitas dan stabilitas aplikasi sebelum dan setelah dipublikasikan.

Testing, Aplikasi, Mobile, Aplikasi Mobile, Blackbox testing, STLC (Software Testing Life Cycle).

I. PENDAHULUAN

Pemberdayaan perempuan menjadi bagian penting dalam pembangunan sosial dan ekonomi di Indonesia, khususnya di tingkat desa. Meskipun memiliki peran signifikan dalam pengambilan keputusan serta pengelolaan sumber daya, partisipasi perempuan sering kali masih belum diakomodasi secara optimal. Data dari Badan Pusat Statistik (BPS, 2019) menunjukkan bahwa keterlibatan perempuan dalam aktivitas ekonomi dan sosial di desa masih rendah, sehingga diperlukan pendekatan inovatif untuk meningkatkan peran aktif mereka dalam pembangunan.

Salah satu bentuk inovasi tersebut adalah melalui pemanfaatan teknologi digital. Aplikasi Sicantik Bangsa dikembangkan sebagai platform yang mendukung pemberdayaan perempuan Indonesia, khususnya dalam mencatat dan memfasilitasi partisipasi mereka dalam.

kegiatan pembangunan desa. Aplikasi ini menyediakan fitur yang ramah pengguna agar perempuan dapat menyampaikan aspirasi, berinovasi, serta berkontribusi secara langsung. Namun, berdasarkan hasil kuesioner yang disebarkan kepada kader pengguna, ditemukan bahwa terdapat beberapa fitur dalam aplikasi yang tidak berfungsi dengan baik. Hal ini mengindikasikan adanya potensi bug atau ketidaksesuaian fitur terhadap kebutuhan pengguna.

Untuk menangani permasalahan tersebut, diperlukan proses pengujian perangkat lunak yang sistematis guna mengidentifikasi akar permasalahan. Pendekatan Software Testing Life Cycle (STLC) dapat digunakan sebagai kerangka kerja pengujian karena mencakup seluruh tahapan pengujian, mulai dari perencanaan hingga pelaporan. Dalam penelitian ini, pengujian dilakukan menggunakan dua metode, yaitu manual testing dengan Qase.io dan automated testing menggunakan Katalon Studio. Manual testing dilakukan berdasarkan skenario tertentu untuk menguji kecocokan fitur dengan kebutuhan fungsional, sedangkan automated testing digunakan untuk mendeteksi regresi dan menguji fitur secara berulang dengan efisiensi tinggi.

Keduanya menerapkan teknik black-box testing yang memfokuskan pengujian pada fungsionalitas aplikasi berdasarkan input dan output, tanpa melihat struktur kode program. Teknik ini terbukti efektif sebagaimana ditunjukkan oleh penelitian Kurnia Gusti Ayu et al. (2024), yang menerapkan STLC dan black-box testing pada aplikasi HESTI. Penelitian tersebut menunjukkan tingkat keberhasilan yang tinggi, dengan fungsionalitas aplikasi mencapai 100% dan performa 90%, serta tidak ditemukannya masalah kompatibilitas.

Berdasarkan hal tersebut, penerapan metode STLC yang dikombinasikan dengan manual dan automated testing dalam penelitian ini diharapkan dapat mengoptimalkan proses pengujian aplikasi Sicantik Bangsa, serta menghasilkan data yang relevan untuk peningkatan kualitas aplikasi sesuai kebutuhan pengguna.

II. KAJIAN TEORI

Kajian teori pada penelitian ini membahas konsep dan dasar teoritis yang mendukung proses perancangan, pengujian, dan analisis aplikasi Sicantik Bangsa. Kajian ini memperkuat metodologi yang digunakan agar sesuai dengan prinsip ilmiah dan praktik di bidang rekayasa perangkat lunak. Teori yang diangkat mencakup optimisasi pengujian, karakteristik aplikasi mobile, pendekatan STLC, black-box testing, serta pemanfaatan alat bantu seperti Katalon Studio, Qase.io, dan emulator Android.

A. Optimisasi

Optimisasi adalah proses mencari solusi terbaik dari suatu permasalahan dengan mempertimbangkan batasan tertentu. Dalam konteks software testing, optimisasi berarti meningkatkan kualitas, cakupan, dan efisiensi pengujian perangkat lunak dengan metode yang terstruktur. Tujuannya adalah memperoleh hasil maksimal dengan sumber daya terbatas. Ammann & Offutt (2016) menyatakan bahwa otomatisasi dalam pengujian, khususnya black-box testing, merupakan strategi penting dalam optimisasi, karena mendukung pelaksanaan pengujian secara cepat dan sistematis sesuai tahapan STLC.

B. Aplikasi Mobile

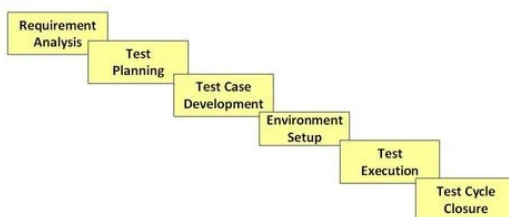
Aplikasi mobile berperan penting dalam kehidupan sehari-hari, menyediakan akses cepat terhadap berbagai layanan. Dalam pengujian perangkat lunak, aplikasi mobile menuntut pendekatan khusus karena melibatkan elemen seperti sensor, layar sentuh, dan koneksi jaringan. Oleh karena itu, proses pengujian pada platform ini memerlukan penyesuaian baik dari sisi metode maupun alat bantu.

C. Sicantik Bangsa

Sicantik Bangsa adalah aplikasi mobile yang dikembangkan untuk mencatat partisipasi perempuan dalam pembangunan desa. Aplikasi ini mendukung program pemberdayaan gender dan telah diimplementasikan dalam program WANI LEMPER di Kabupaten Kebumen, bekerja sama dengan Institut Teknologi Telkom Purwokerto.

D. STLC (Software Testing Life Cycle)

STLC merupakan metodologi pengujian perangkat lunak yang mencakup tahapan dari perencanaan hingga pelaporan. Berbeda dari SDLC, STLC fokus pada proses pengujian untuk memastikan kualitas sistem. Penerapan STLC membantu meningkatkan efektivitas, efisiensi, dan keandalan hasil uji.



GAMBAR 1

ALUR STLC (SUMBER: GURU99.COM)

E. Black Box Testing

Black-box testing adalah metode pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem berdasarkan input dan output, tanpa mengetahui struktur internalnya. Pendekatan ini berguna untuk memvalidasi bahwa sistem bekerja sesuai dengan kebutuhan pengguna.

F. Katalon

Katalon adalah alat otomatisasi pengujian yang mendukung berbagai jenis pengujian seperti fungsional, API, dan mobile. Katalon membantu tim pengembang mempercepat proses pengujian dan meningkatkan kualitas perangkat lunak dengan antarmuka yang mudah digunakan.

G. Qase.io

Qase.io adalah platform manajemen pengujian yang mendukung pengelolaan test case, pelacakan bug, dan integrasi dengan alat pengembangan lain. Dengan antarmuka intuitif, Qase.io membantu tim meningkatkan kolaborasi dan efisiensi dalam proses pengujian.

H. Bluestack

Bluestack adalah emulator Android yang memungkinkan aplikasi mobile dijalankan di PC. Versi terbaru seperti Bluestack 5 menawarkan peningkatan performa dan stabilitas, sehingga banyak digunakan oleh pengembang dan pengguna umum untuk pengujian aplikasi.

I. AVD Android Studio

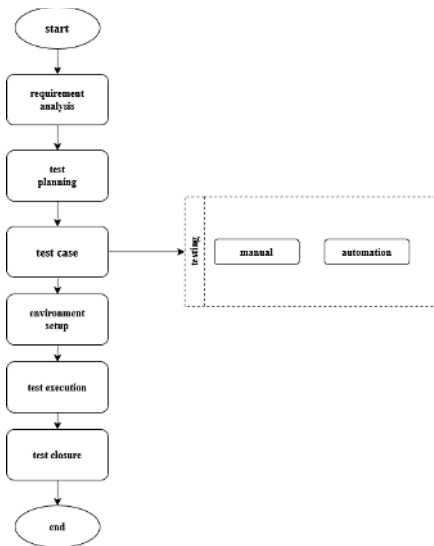
Android Virtual Device (AVD) adalah emulator dari Android Studio yang memungkinkan pengujian aplikasi pada berbagai perangkat virtual. AVD berguna untuk menguji aplikasi dalam berbagai konfigurasi perangkat dan sistem operasi tanpa perangkat fisik.

III. METODE

Penelitian ini menggunakan pendekatan studi kasus dengan metode Software Testing Life Cycle (STLC) sebagai kerangka utama dalam proses pengujian aplikasi Sicantik Bangsa. STLC dipilih karena menyediakan alur pengujian yang terstruktur dan sistematis mulai dari tahap perencanaan hingga pelaporan hasil. Pengujian dilakukan terhadap aplikasi Sicantik Bangsa dengan dua pendekatan, yaitu manual dan otomatisasi, menggunakan platform Qase.io dan Katalon Studio.

A. Diagram Alir Penelitian

Definisikan singkatan dan akronim saat pertama kali digunakan dalam teks, bahkan setelah didefinisikan dalam abstrak. Jangan menggunakan singkatan dalam judul kecuali jika tidak dapat dihindari.



GAMBAR 2
DIAGRAM ALIR PENELITIAN

Berdasarkan Gambar 1.2, proses penelitian dilakukan mengikuti tahapan-tahapan dalam STLC yang mencakup: Requirement Analysis, Test Planning, Test Case Development, Environment Setup, Test Execution, dan Test Closure. Penjelasan rinci dari tiap tahap dijabarkan dalam subbab berikut:

B. Requirement Analysis

Tahap awal ini bertujuan untuk menganalisis kebutuhan fungsional aplikasi berdasarkan hak akses pengguna, yaitu Koordinator, Kader, dan Masyarakat. Pengujian difokuskan pada fitur-fitur utama menggunakan pendekatan Black Box Testing yang tidak memerlukan pemahaman terhadap kode program, melainkan fokus pada fungsi aplikasi.

C. Test Planning

Perencanaan pengujian mencakup fitur yang akan diuji dan strategi pelaksanaannya. Rencana ini memastikan seluruh kebutuhan pengguna tercakup agar pengujian dapat dilakukan secara optimal.

D. Test Case Development

Test case disusun berdasarkan rencana yang telah ditentukan sebelumnya dengan pendekatan modularisasi, yaitu:

- Modul fungsional aplikasi
- Jenis skenario (positif dan negatif)
- Hak akses pengguna (role-based)

Pendekatan ini bertujuan mempermudah manajemen dan pelaksanaan pengujian serta pelacakan bug.

E. Test Environment Setup

Persiapan dilakukan terhadap perangkat keras dan lunak yang diperlukan. Pengujian menggunakan Windows 11, Bluestacks 5, AVD (Android Virtual Device), Katalon Studio 9.1.0, Qase.io, dan Mozilla Firefox.

F. Test Execution

Eksekusi dilakukan terhadap seluruh test case, baik secara manual (Qase.io) maupun otomatis (Katalon Studio). Status hasil ditandai secara real-time (hijau untuk pass, merah untuk fail). Test suite disusun berdasarkan role pengguna agar hasil lebih terstruktur.

G. Test Closure

Tahap akhir mencakup evaluasi menyeluruh terhadap proses pengujian. Kegiatan meliputi penyusunan laporan, dokumentasi hasil eksekusi, analisis defect, dan peninjauan ketercapaian exit criteria. Evaluasi juga dilakukan sebagai dasar peningkatan proses pengujian ke depan.

IV. HASIL DAN PEMBAHASAN

Bagian ini berisi paparan objektif peneliti terhadap hasil-hasil penelitian berupa penjelasan dan analisis terhadap penemuan-penemuan penelitian, penjelasan serta penafsiran dari data dan hubungan yang diperoleh, serta pembuatan generalisasi dari penemuan. Apabila terdapat hipotesis, maka pada bagian ini juga menjelaskan proses pengujian hipotesis beserta hasilnya.

Hasil penelitian harus disajikan secara jelas dan sistematis supaya mudah dibaca dan dipahami. Penyajian hasil penelitian dapat dilakukan dengan cara deskriptif (naratif), menggunakan tabulasi, tabel atau grafik, atau dengan menggunakan gabungan dua atau ketiganya secara sekaligus. Penggunaan ketiga cara tersebut disesuaikan dengan jenis data dan sejauh mana deskripsi data akan dijelaskan. Misalnya, pada awal peneliti memaparkan narasi temuannya, kemudian didukung dengan sajian data dalam bentuk tabulasi, tabel atau grafik. Peneliti juga menyajikan data-data hasil penelitian, kemudian didukung grafik dilanjutkan deskripsi naratif [10 pts]. Berikan kemungkinan pengembangan atau penelitian ke depan terkait penelitian ini

A. Requirement Analysis

Pada tahap ini dilakukan analisis terhadap kebutuhan fungsional dan platform Sicantik Bangsa. Kebutuhan tersebut mencakup berbagai fitur utama. Pengujian difokuskan pada aspek fungsionalitas tanpa memerlukan pemahaman mendalam tentang kode sumber aplikasi, dengan menggunakan metode Black Box Testing. Adapun fitur-fitur yang diuji didasarkan pada hak akses masing-masing role pengguna, sebagaimana ditunjukkan pada tabel 3.1 dibawah ini.

TABEL 1
REQUIREMENT ANALYSIS

No	Fitur	Koordinator	Kader	Masyarakat
1	Autentikasi	Y	Y	Y
2	Dashboard	Y	Y	Y
3	Mengelola Aspirasi	Y	Y	N
4	Menambah Aspirasi	Y	Y	Y
5	Mengelola dan Menambah Usulan	Y	Y	N
6	Notulensi	Y	Y	N
7	Verifikasi pengguna	Y	N	N
8	Pengaturan	Y	Y	Y

B. Test Planning

Pengujian aplikasi Sicantik Bangsa dilakukan dengan menggunakan metode Software Testing Life Cycle (STLC) untuk memastikan bahwa seluruh fitur dalam aplikasi berfungsi dengan baik dan sesuai dengan kebutuhan pengguna.

berikut daftar fitur utama yang akan menjadi fokus pengujian adalah sebagai berikut:

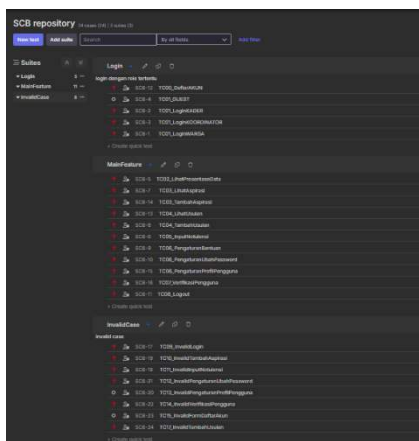
- Fitur Autentikasi dapat berfungsi dengan baik
- Fitur Dashboard dapat berfungsi dengan baik
- Fitur Aspirasi dapat berfungsi dengan baik
- Fitur Usulan dapat berfungsi dengan baik
- Fitur Notulensi dapat berfungsi dengan baik
- Fitur Verifikasi dapat berfungsi dengan baik
- Fitur Pengaturan dapat berfungsi dengan baik

C. Test Case

Pada tahap ini, dilakukan penyusunan test case dan test scenario yang merujuk pada test planning yang telah dirancang sebelumnya.

• Test Case Manual

Pembuatan Test case manual dilakukan dengan bantuan qase.io untuk memudahkan dokumentasi dan penyusunan scenario

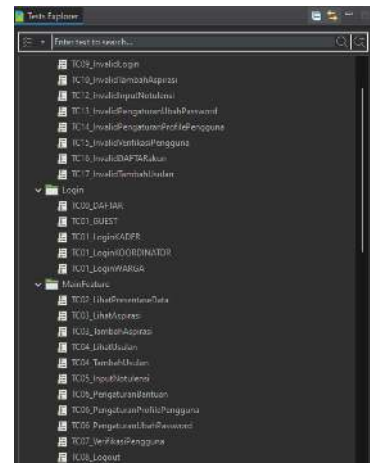


GAMBAR 3
TEST CASE MANUAL QASE IO

Gambar 3 menampilkan rancangan test case manual. Test case dikelompokkan ke dalam Test suite berdasarkan fitur utama aplikasi dan Invalid case untuk mempermudah pembuatan Test runs.

• Test Case Automation

Pembuatan Test case automation dilakukan dengan memanfaatkan fitur Record Mobile untuk mengenali dan get elemen pada aplikasi, Lalu skrip test case dibuat secara manual yang scenarionya mengacu pada pembuatan test case manual yang sebelumnya sudah dibuat.



GAMBAR 4
TEST CASE AUTOMATION KATALON

Gambar 4 diatas menunjukkan struktur test case yang digunakan dalam pengujian otomatis. Setiap test case dinamai secara sistematis menggunakan prefix dan deskripsi singkat untuk memudahkan identifikasi serta pemetaan skenario otomatis yang relevan dengan pengujian.

D. Environment Setup

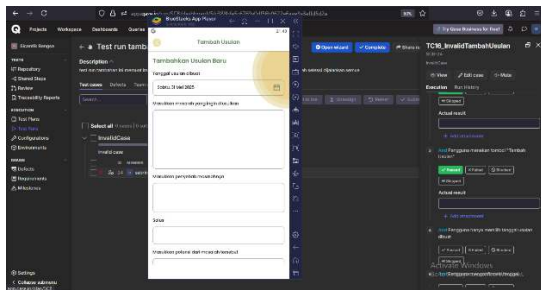
Pada tahap ini, dilakukan persiapan untuk memenuhi syarat lingkungan pengujian yang mencakup seluruh konfigurasi teknis yang diperlukan sebelum pelaksanaan pengujian. Persyaratan ini meliputi spesifikasi perangkat keras dan perangkat lunak yang memadai, seperti sistem operasi Windows 11 64-bit dengan RAM minimal 4GB. Selain itu, diperlukan juga aplikasi Bluestack, Katalon Studio versi 9.1.0, Mozilla Firefox, dan Qase.io. Dengan memenuhi semua persyaratan ini, diharapkan pengujian dapat berjalan dengan lancar.

E. Test Execution

Tahap pelaksanaan pengujian (test execution) merupakan fase implementasi dari rencana pengujian yang telah disusun sebelumnya. Pada tahap ini dilakukan eksekusi terhadap seluruh skenario pengujian.

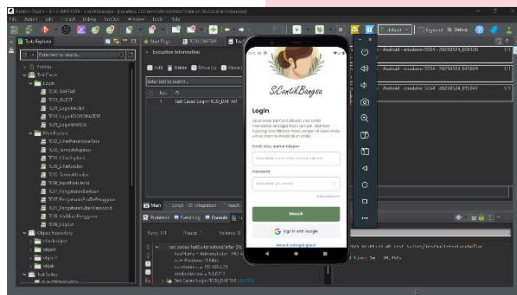
Proses eksekusi pengujian dilakukan dengan pendekatan sistematis dimana test suite diorganisir berdasarkan peran pengguna (role-based testing) untuk memastikan setiap fungsi sistem dapat diuji sesuai dengan hak akses dan tanggung jawab masing-masing role. Pemisahan test suite berdasarkan role ini bertujuan untuk menghasilkan laporan pengujian yang lebih terstruktur dan komprehensif, sehingga memudahkan dalam mengidentifikasi masalah yang spesifik untuk setiap kategori pengguna.

Pada proses ini, hasil eksekusi test case ditandai dengan warna tertentu sebagai indikator status pengujian. Penandaan berwarna merah menunjukkan bahwa test case berstatus failed, sedangkan warna hijau menandakan bahwa test case telah pass. Contoh hasil test execution dapat dilihat pada Gambar 3.3.6 dan Gambar 3.3.7 di bawah ini.



GAMBAR 5
TEST CASE EXECUTION MANUAL

Gambar 5 menunjukkan contoh proses *execution* pada fitur *Test Runs* di Qase.io, di mana proses eksekusi dibantu dengan menggunakan emulator Bluestacks 5. Pada *Test Run* tersebut, waktu pelaksanaan tercatat secara *real-time* selama proses pengujian berlangsung.



GAMBAR 6
TEST CASE EXECUTION AUTOMATION

Gambar 6 menunjukkan salah satu contoh proses *execution automation* pada test case menggunakan Katalon Studio yang dijalankan pada perangkat virtual Android (AVD). Dalam proses ini, *test case* berjalan secara otomatis berdasarkan skrip *automation* yang telah disusun sebelumnya. Waktu pelaksanaan pengujian dicatat secara *real-time* oleh sistem, dimulai sejak aplikasi dijalankan.

F. Test Closure

Tahap penutupan siklus pengujian merupakan fase akhir dari proses pengujian perangkat lunak yang bertujuan untuk mengevaluasi kelengkapan dan hasil seluruh aktivitas pengujian yang telah dilaksanakan. Pada tahap ini dilakukan kompilasi dan analisis menyeluruh terhadap semua hasil pengujian, dokumentasi temuan, serta evaluasi pencapaian kriteria keluar (exit criteria) yang telah ditetapkan sebelumnya.

G. Implementasi Hasil Pengujian

Pengujian dilakukan terhadap seluruh fitur utama pada aplikasi Sicantik Bangsa melalui pendekatan manual dan otomatis. Pengujian manual dilakukan dengan Qase.io, sedangkan pengujian otomatis dilakukan dengan Katalon Studio.

• Manual Testing

Manual testing menghasilkan 46 skenario uji. Hasil pengujian mencatat total 20 bug dengan rincian:



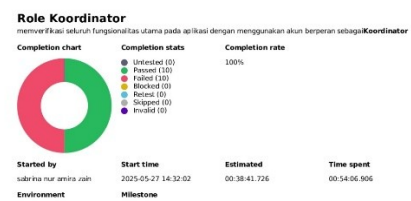
GAMBAR 7
HASIL TEST RUN WARGA

Pada Gambar 7 ditampilkan hasil keseluruhan hasil uji dengan role warga dengan mementingkan aspek fungsionalitas utama pada aplikasi ketika pengguna masuk sebagai role warga. Dari total 11 test case yang dijalankan 7 test case berstatus passed dan 4 test case berstatus failed. Pengujian ini dilakukan dengan waktu pelaksanaan 29 menit 56.628 Detik.



GAMBAR 8
HASIL TEST RUN ROLE KADER

Pada Gambar 8 ditampilkan hasil keseluruhan hasil uji dengan role kader dengan mementingkan aspek fungsionalitas utama pada aplikasi ketika pengguna masuk sebagai role kader. Dari total 15 test case yang dijalankan 10 test case berstatus passed dan 5 test case berstatus failed. Pengujian ini dilakukan dengan waktu pelaksanaan 33 menit 25.509 Detik.



GAMBAR 9
HASIL TEST RUN ROLE KOORDINATOR

Pada Gambar 9 ditampilkan hasil keseluruhan hasil uji dengan role koordinator dengan mementingkan aspek fungsionalitas utama pada aplikasi ketika pengguna masuk sebagai role koordinator. Dari total 20 test case yang dijalankan 10 test case berstatus passed dan 10 test case berstatus failed. Pengujian ini dilakukan dengan waktu pelaksanaan 54 menit 06.906 Detik.



GAMBAR 10
HASIL TEST RUN TAMBAHAN

Pada Gambar 10 ditampilkan hasil keseluruhan hasil uji kondisi invalid dengan mementingkan aspek fungsionalitas utama pada aplikasi ketika pengguna masuk sebagai role kader pada gambar 4.49 dan koordinator pada gambar 4.50. Dari test case yang dijalankan semuanya berstatus passed. Pengujian ini dilakukan dengan waktu pelaksanaan 12 menit 44.322 Detik.

- Automation Testing

Automation testing menghasilkan 46 skenario uji yang sama. Hasil eksekusi menunjukkan 9 test case failed akibat kesalahan sistem atau UI.

GAMBAR 11
TEST RUN SUITE ROLE WARGA

Gambar 11 menunjukkan hasil test run otomatis pada Test Suite Role Warga yang dijalankan menggunakan Katalon Studio. Dari 11 test case yang diuji, 10 dinyatakan Passed, dan 1 Failed, yaitu PengaturanProfilePengguna.

GAMBAR 12
TEST RUN SUITE ROLE KADER

Gambar 12 menunjukkan hasil *test run* otomatis pada Test Suite Role KADER yang dijalankan menggunakan Katalon Studio. Dari total 16 test case yang diuji, 11 dinyatakan Passed dan 5 Failed.

GAMBAR 13
TEST RUN SUITE ROLE KOORDINATOR

Gambar 13 menunjukkan hasil test run otomatis pada Test Suite Role KOORDINATOR yang dijalankan menggunakan Katalon Studio. Dari total 18 test case yang diuji, sebanyak 12 dinyatakan Passed dan 6 Failed.

H. Analisis Hasil Pengujian

Berikut dilakukan analisis terhadap hasil pengujian antara dua metode testing, dengan fokus pada dua metrik utama yaitu efisiensi waktu dan tingkat deteksi bug.

- Efisiensi Waktu

Efisiensi waktu dihitung dengan membandingkan durasi pelaksanaan pengujian antara dua metode sebagai berikut :

$$Time\ Efficiency = \frac{Waktu\ Manual\ Testing}{Waktu\ Automation\ Testing}$$

Interpretasi : Jika hasil > 1, maka automation lebih efisiensi dari sisi waktu.

Maka jika berdasarkan rumus diatas time efficiency, didapatkan:

Spend time manual testing :

- Role Warga : 29 menit 56.628 detik
- Role Kader : 44 menit 14.765 detik
- Role Koordinator : 56 menit 01.972 detik

Total : 29:56.628 + 44:14.765 + 56:01.972

Total Waktu = 2 jam 10 menit 13.365 detik

= (2 × 3600) + (10 × 60) + 13.365

= 7200 + 600 + 13.365

= 7813.365 detik

Spend time automation testing :

- Role Warga : 3 menit 37.750 detik
- Role Kader : 4 menit 39.207 detik
- Role Koordinator : 4 menit 44.328 detik

Total : 03:37.750 + 04:39.207 + 04:44.328

Total Waktu = 13 menit 01.285 detik

13 menit 01.285 detik

= (13 × 60) + 1.285

= 780 + 1.285

= 781.285 detik

$$Time\ Efficiency = \frac{7813.365}{781.285} \approx 10.0$$

Maka dapat disimpulkan bahwa automation testing memiliki efisiensi waktu sekitar 10 kali lebih cepat dibandingkan dengan manual testing. Hal ini menunjukkan bahwa automation lebih unggul dari segi efisiensi waktu.

- Tingkat Deteksi Bug (Bug Detection Rate)
BDR merupakan ukuran efektivitas metode pengujian dalam menemukan bug, dihitung dengan rumus:

$$\text{BDR} = \frac{\text{Jumlah Bug Ditemukan}}{\text{Total Skenario Uji}} \times 100\%$$

Nilai BDR yang lebih tinggi menunjukkan bahwa metode pengujian tersebut mampu mendeteksi bug dengan lebih efektif.

Maka jika berdasarkan rumus BDR diatas, didapatkan :

BDR manual testing :

$$\text{BDR} = \frac{20}{46} \times 100\% = 43.478\%$$

BDR automation testing :

$$\text{BDR} = \frac{9}{46} \times 100\% = 19.57\%$$

V. KESIMPULAN

Berdasarkan hasil implementasi pengujian aplikasi Sicantik Bangsa menggunakan pendekatan kombinasi antara manual testing dan automation testing yang terintegrasi dalam Software Testing Life Cycle (STLC), dapat disimpulkan bahwa pengujian otomatis mampu mempercepat proses eksekusi dibandingkan metode konvensional. Hal ini mendukung percepatan siklus pengembangan dan memungkinkan identifikasi bug secara lebih dini. Sementara itu, pengujian manual menunjukkan kemampuan yang lebih kuat dalam mendeteksi kesalahan sistem yang bersifat kompleks, termasuk isu yang berkaitan dengan alur logika dan pengalaman pengguna, yang sering kali tidak terjangkau oleh skrip otomatis.

Dengan menggabungkan kedua pendekatan ini, strategi pengujian menjadi lebih optimal karena mampu menyeimbangkan kecepatan dan ketelitian, sekaligus mengurangi kemungkinan false positive dan false negative. Penerapan STLC menjadikan proses pengujian lebih terstruktur dan terdokumentasi dengan baik, serta dapat mengoptimalkan kualitas pengujian dan diadopsi dalam proyek pengembangan aplikasi serupa.

REFERENSI

- [1] R. A. Sari and R. Sulastri, "Pemberdayaan Perempuan dalam Pembangunan Ekonomi di Karawang," *Gunung Djati Conference Series*, vol. 39, pp. 45–53, 2024.
- [2] A. R. Bahtiar *et al.*, "Implementasi platform Si Cantik Bangsa di Kabupaten Kebumen dalam pembangunan desa," *Warta LPM*, vol. 27, no. 2, pp. 380–391, 2024, doi: 10.23917/warta.v27i2.4815.
- [3] A. Arfan and Hendrik, "Penerapan STLC dalam pengujian automation aplikasi mobile (studi kasus: LMS Amikom Center PT. GIT Solution)," *Informatika UIH*, vol. 3, no. 2, pp. 1–6, 2022.
- [4] K. G. Ayu, D. W. Sari, and R. S. Fadila, "CESS Pengujian Black-Box Dan Analisa Kualitas ISO 25010 Pada Aplikasi HESTI Black-Box Testing and ISO 25010 Quality Analysis on the HESTI Application," vol. 9, no. 2, pp. 710–721, 2024.
- [5] Y. D. Leksanti, "Pengujian Website ACC Whistle Menggunakan Metode Black Box Testing Program Studi Informatika," p. 185, 2020.
- [6] B. Huda and A. Lia Hananto, "Penerapan Software Testing Life Cycle Pada Pengujian Otomatisasi Platform Dzikra," *CSRID Journal*, vol. 15, no. 1, pp. 1–11, 2023.
- [7] Y. F. Putri, "Automation Regression Testing Pada Aplikasi Teman Diabetes dengan Menggunakan Metode Blackbox Testing," *Program Studi Sistem Informasi Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta*, 2020.
- [8] A. Zulianto, A. Purbasari, N. Suryani, A. I. Susanti, F. R. Rinawan, and W. G. Purnama, "Pemanfaatan Katalon Studio untuk Otomatisasi Pengujian Black-Box pada Aplikasi iPosyandu," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 7, no. 3, p. 370, 2021, doi: 10.26418/jp.v7i3.46954.
- [9] F. W. Azhari and D. F. Suyatno, "Pengujian Otomatis GUI dengan Katalon Studio pada Situs Lowongan Kerja Jobstreet dan Glints," vol. 05, no. 03, pp. 205–213, 2024.
- [10] H. Judul, F. T. Industri, and U. I. Indonesia, "Pengujian Otomatis Aplikasi Mobile Dengan Teknik Black-Box Menggunakan Appium Pe N Gujia N Otomatis Aplikasi Mobile De N Ga N Tek N Ik Black-Box Me N Ggu N Aka N Appium," 2022.
- [11] S. Yasmine Shalsabilla, E. Dyar Wahyuni, and N. Cahyo Wibowo, "Implementasi Blackbox Automation Testing Pada Aplikasi Donor Menggunakan Framework Stlc Dalam Lingkup Pengembangan Agile Scrum," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 2261–2269, 2024, doi: 10.36040/jati.v8i2.9465.
- [12] T. Desyani, A. Syamsiana, E. U. Kobun, F. T. Ananda, S. Priaji, and Y. Yulianti, "Otomatisasi Pengujian Aplikasi Web Otten Coffee Menggunakan Katalon Studio," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 2, pp. 186–190, 2023, doi: 10.32493/jtsi.v6i2.26902.
- [13] P. Mulyah *et al.*, "Exploring learners' autonomy in online language learning in STAI Sufyan Tsauri Majenang," *Getsempena English Educ. J.*, vol. 7, no. 2, pp. 382–394, Nov. 2020, doi: 10.46244/geej.v7i2.1164.
- [14] T. O. Teori, "Teknik Optimasi (Teori, Konsep, dan Aplikasi)," no. April, 2024.
- [15] P. Ammann and J. Offutt, "Introduction To Software Testing," *Introduction to Software Testing*, no. July, pp. 1–345, 2016, doi: 10.1017/9781316771273.
- [16] S. Ziha Fidela, M. Putri Azizah, and S. Rizka Hidayah, "Tren Pengembangan Aplikasi Mobile: Sebuah Tinjauan Literatur," *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika (Jtmei)*, vol. 2, no. 4, pp. 30–48, 2023.
- [17] A. Anand and A. Uddin, "Importance of Software Testing in the Process of Software Development," *IJSRD-International Journal for Scientific Research*

- & Development], vol. 6, no. February, pp. 232-240, 2019.
- [18] V. T. Tempomona, "Penerapan Metode Blackbox Pada Perangkat Lunak Menggunakan Kata Kunci," *INO/TEK Polbeng - Seri Informatika*, vol. 7, no. 2, p. 193, 2022, doi: 10.35314/isi.v7i2.2513.
- [19] S. Laila Ramadhani and H. Prihantoro Putro, "Manajemen Pengujian Perangkat Lunak Menggunakan Aplikasi Qase," *Automata*, vol. 4, no. 1, 2023.
- Genymotion plugin for Android Studio*, Genymobile, Jun. 2025. [Online]. Available: <https://docs.genymotion.com/plugins/android-studio/>
- D. Y. Sylfania, F. P. Juniawan, and D. Yulanda, "Implementasi Aplikasi Pemesanan Makanan Dan Minuman Pada Cafe La Banca Berbasis Android," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 4, no. 2, p. 144, 2021, doi: 10.37600/tekinkom.v4i2.318.

