

Perbandingan Efektivitas Pengujian Manual dan Otomatis Menggunakan System Testing dengan Pendekatan AHP

KURNIA RAMADHAN PUTRA*, MOCHAMAD FAQIH FAIZAL, SOFIA UMAROH

Program Studi Sistem Informasi, Institut Teknologi Nasional Bandung, Indonesia

Email: kurniaramadhan@itenas.ac.id

Received 2 Mei 2026 | Revised 2 Juni 2026 | Accepted 29 Juni 2026

ABSTRAK

Pemilihan strategi pengujian seringkali subjektif tanpa mempertimbangkan karakteristik sistem. Penelitian ini mengevaluasi efektivitas pengujian manual dan otomatis menggunakan Analytic Hierarchy Process (AHP) pada studi kasus SIM SKK dan RPL untuk mengatasi subjektivitas pemilihan metode uji dan menentukan prioritas kriteria yang digunakan untuk membandingkan pengujian manual dan otomatis tersebut. Evaluasi diukur secara objektif melalui teknik Checklist-based Testing berdasarkan empat kriteria utama: waktu pengujian, cakupan pengujian, biaya pengujian berbasis Time-Driven Activity-Based Costing (TDABC), dan kuantitas penemuan bug. Pengujian otomatis menggunakan Katalon Studio terbukti menjadi strategi paling optimal pada kedua studi kasus karena mendominasi kriteria penemuan bug (33 pada SIM SKK dan 40 pada RPL) serta efisiensi waktu, sehingga menghasilkan Skor Global yang mengungguli metode manual baik pada SIM SKK (0,9697 berbanding 0,8986) maupun Sistem RPL (0,9379 berbanding 0,9002). Pengujian otomatis unggul karena dominasi mutlak pada aspek penemuan bug dan kecepatan eksekusi yang memiliki bobot kepentingan tertinggi dalam model keputusan AHP.

Kata kunci: *Software Quality, Automated Testing, Manual Testing, TDABC, AHP*

ABSTRACT

The selection of testing strategies is often subjective and may overlook system characteristics. This study evaluates manual and automated testing using the Analytic Hierarchy Process (AHP) on SIM SKK and RPL systems. Evaluation was conducted through Checklist-based Testing based on four criteria: testing time, test coverage, testing cost, and bug detection. Automated testing using Katalon Studio proved to be the most effective approach due to better bug detection and higher time efficiency. As a result, automated testing achieved higher Global Scores than manual testing in both SIM SKK (0.9697 vs. 0.8986) and RPL (0.9379 vs. 0.9002). Its superiority is mainly attributed to bug detection capability and execution speed, which received the highest weights in the AHP model.

Keywords: *Software Quality, Automated Testing, Manual Testing, TDABC, AHP*

1. PENDAHULUAN

Kualitas perangkat lunak menjadi instrumen krusial untuk mencegah risiko kegagalan sistem yang berakibat pada kerugian finansial maupun penurunan reputasi penyedia layanan (**Putri et al., 2025**). Salah satu upaya yang dapat dilakukan untuk mencegah risiko kegagalan sistem tersebut serta meningkatkan kualitas perangkat lunak yang dihasilkan, maka diperlukan pengujian perangkat lunak yang merupakan salah satu fase di dalam kerangka kerja *Software Development Life Cycle* (SDLC), mulai tahap perencanaan, perancangan, penerapan, pengujian, hingga pemeliharaan (**Hossain, 2023; Huda et al., 2023**). Di antara seluruh tahapan tersebut, pengujian perangkat lunak merupakan elemen krusial yang berfungsi sebagai instrumen penjamin kualitas dengan tujuan utama mengidentifikasi kesalahan dan celah keamanan sebelum sistem dirilis kepada pengguna (**Honest, 2019**). Tanpa proses pengujian yang ketat, risiko kegagalan sistem dapat berakibat pada kerugian finansial maupun penurunan reputasi penyedia layanan.

Secara operasional, pengujian perangkat lunak dikategorikan menjadi dua metode utama, yaitu manual dan otomatis (**Thant & Tin, 2023**). Pengujian manual sangat bergantung pada ketelitian penguji dalam menjalankan skenario uji secara langkah demi langkah, yang memberikan fleksibilitas tinggi dalam menilai aspek intuitif dan risiko sistem secara mendalam (**Min et al., 2020; Riski & Renanti, 2024**). Namun, metode tersebut sering kali memicu kesalahan faktor manusia yang tidak disengaja, sehingga membutuhkan sumber daya manusia yang besar, serta membutuhkan waktu eksekusi yang signifikan (**Ahmad & Hinchey, 2023; Zafar & Enoiu, 2022**). Sebaliknya, pengujian otomatis menggunakan skrip dan alat bantu otomatis yang mampu mempercepat proses eksekusi hingga 70%, meningkatkan konsistensi, serta memudahkan *regression testing* (pengulangan pengujian) (**Kumari et al., 2018**). Meskipun demikian, otomatisasi menuntut biaya investasi awal yang tinggi dan keterampilan teknis dalam pemeliharaan skrip uji agar tetap relevan dengan perubahan sistem.

Meskipun perbandingan antara pengujian manual dan otomatis telah banyak dibahas dalam literatur, sebagian besar evaluasi yang dilakukan masih bersifat parsial atau kualitatif (**Ardic et al., 2025; Yusuf, 2022**). Penentuan efektivitas sering kali hanya menitikberatkan pada satu aspek, seperti kecepatan eksekusi atau biaya saja, tanpa mempertimbangkan *trade-off* (kompromi) antar kriteria secara komprehensif. Terdapat celah penelitian dalam hal standarisasi pengukuran yang mampu mengintegrasikan berbagai kriteria kuantitatif dan kualitatif ke dalam satu model pengambilan keputusan yang terukur. Tanpa pendekatan multi-kriteria, keputusan pemilihan metode uji berisiko menjadi subjektif dan tidak optimal bagi kebutuhan proyek tertentu.

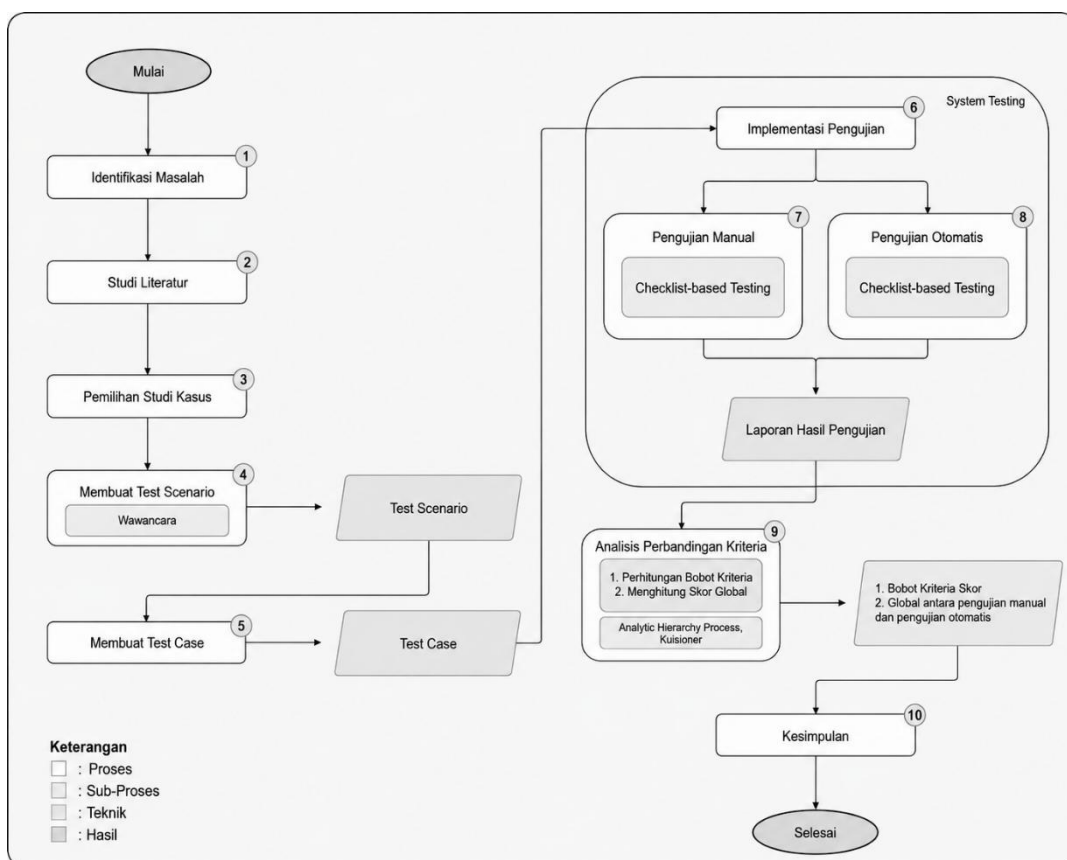
Masalah utama dalam pemilihan metode pengujian bukan sekadar pada kecepatan eksekusi, melainkan pada bagaimana sifat alur kerja sistem menentukan efektivitas alat uji. Urgensi penelitian ini ditekankan pada *System Testing* menggunakan dua studi kasus yang berbeda karakteristik, yaitu Sistem Informasi Manajemen Satuan Kredit Kemahasiswaan (SIM SKK) dan Sistem Sistem Rekognisi Pembelajaran Lampau (RPL). Pada sistem dengan alur kerja dinamis seperti SIM SKK, yang melibatkan verifikasi bertingkat dan orientasi status (poin), skrip otomatisasi dapat mengeksplorasi efisiensi waktu secara penuh tanpa mengalami kendala penurunan cakupan pengujian. Sebaliknya, pada sistem dengan alur kerja linear dan kaku seperti Sistem RPL, terdapat tantangan berupa kerapuhan skrip. Karakteristik linear yang memiliki ketergantungan prosedur yang sangat ketat dapat menyebabkan skrip otomatis rentan berhenti jika terjadi anomali kecil, yang berakibat pada penurunan cakupan uji. Pemilihan kedua sistem ini didasarkan pada variasi kompleksitas alur kerja dan integritas data yang dikelola. SIM SKK lebih berfokus pada manajemen data aktivitas mahasiswa yang masif,

sementara RPL melibatkan validasi dokumen dan aturan konvergensi nilai yang lebih kompleks.

Sebagai solusi untuk mengatasi subjektivitas dalam evaluasi tersebut, penelitian ini menerapkan metode *Analytic Hierarchy Process* (AHP) dengan tujuan untuk menghasilkan perbandingan efektivitas pengujian yang objektif melalui tiga luaran spesifik, yaitu matriks pembobotan prioritas, model skor global, dan landasan pengambilan keputusan yang dapat mereduksi bias subjektivitas dalam memilih strategi pengujian yang paling optimal bagi kualitas sistem. AHP memungkinkan penilaian faktor-faktor efektivitas seperti waktu pengujian, biaya pengujian, cakupan pengujian, dan penemuan *bug* secara terstruktur melalui pembobotan matematis dan analisis konsistensi (Abdulwareth & Al-shargabi, 2021; Zarrad et al., 2024). Validitas ilmiah dalam penentuan bobot kriteria dijamin melalui perhitungan Consistency Ratio (CR) untuk memastikan tidak adanya bias atau kontradiksi dalam penilaian para pakar. Data penilaian responden dinyatakan valid konsisten secara ilmiah jika memenuhi ambang batas $CR < 0,1$ (Rochimah et al., 2023). Hasil akhir dari penelitian ini dapat digunakan sebagai acuan dalam pertimbangan pengembang perangkat lunak untuk menentukan metode pengujian yang paling tepat guna menjamin kualitas sistem secara efisien.

2. METODE PENELITIAN

Seluruh tahapan penelitian, mulai dari identifikasi awal hingga penentuan keputusan akhir menggunakan AHP digambarkan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Identifikasi Masalah

Tahap pertama melibatkan identifikasi variabel kritis yang memengaruhi efektivitas pengujian. Permasalahan yang diangkat berfokus pada keterbatasan sumber daya dan kebutuhan akan akurasi dalam siklus pengembangan perangkat lunak. Kriteria yang diidentifikasi sebagai parameter pembandingan adalah:

- a. *Time* atau Waktu Pengujian yaitu durasi yang dibutuhkan untuk menyelesaikan seluruh test case.
- b. *Coverage* atau Cakupan Pengujian yaitu sejauh mana fitur sistem terakomodasi dalam pengujian.
- c. *Cost* atau Biaya Pengujian yaitu biaya yang dikeluarkan untuk investasi sumber daya manusia dan perangkat lunak yang diperlukan.
- d. Penemuan *Bug* atau Defek yaitu efikasi metode dalam mengidentifikasi defek pada sistem.

Setelah variabel kritis seperti waktu, cakupan, biaya, dan bug diidentifikasi pada tahap awal, proses dilanjutkan dengan studi literatur guna membangun landasan teoritis dan memetakan celah penelitian yang mendukung urgensi penggunaan kriteria tersebut dalam pengujian perangkat lunak.

2.2. Studi Literatur

Tahap studi literatur dilakukan untuk membangun landasan teoretis yang kokoh serta memetakan *research gap* (celah penelitian). Penelusuran literatur dilakukan secara sistematis melalui basis data akademik seperti Google Scholar, IEEE Xplore, dan ScienceDirect. Untuk memastikan relevansi sumber, dilakukan klasifikasi referensi berdasarkan kata kunci dan topik utama yang berkaitan dengan variabel penelitian. Ringkasan distribusi referensi yang digunakan dalam penelitian ini disajikan pada Tabel 1.

Tabel 1. Pemetaan Referensi dan Topik Literatur

No.	Topik Utama	Kata Kunci Pencarian	Estimasi Jumlah Referensi
1	Software Testing	System Testing, Blackbox Testing, Manual vs Automated Testing	8-10
2	Testing Efficiency	Test Execution Time, Defect Leakage Rate, Resources per Test, Activity-Based Costing	5-7
3	Decision Making	AHP Method, Multi-criteria Decision Making, Saaty Scale	6-8
4	Automation Tools	Katalon Studio, Selenium, Scripting efficiency	4-5

2.3. Pemilihan Studi Kasus

Pemilihan objek penelitian dilakukan menggunakan teknik *purposive sampling* dengan menetapkan dua sistem informasi yang memiliki karakteristik arsitektur data, hak akses aktor, dan alur kerja yang kontras. Perbedaan karakteristik ini menjadi landasan kuat untuk menguji efektivitas metode pengujian manual dan otomatis dalam menangani variabilitas beban kerja yang berbeda.

Sistem Informasi Manajemen Satuan Kredit Kemahasiswaan (SIM SKK): SIM SKK dipilih untuk merepresentasikan sistem dengan karakteristik manajemen dokumen intensif dan alur kerja dinamis. Sistem berbasis web ini digunakan untuk mendata kegiatan kurikuler dan ekstrakurikuler mahasiswa melalui proses verifikasi bertingkat yang melibatkan aktor Mahasiswa, Dosen Wali, dan Ketua Program Studi.

Sistem Rekognisi Pembelajaran Lampau (Sistem RPL): Sistem RPL dipilih untuk merepresentasikan sistem dengan karakteristik densitas input teks tinggi dan alur kerja linear. Platform ini berfungsi untuk memfasilitasi calon mahasiswa dalam mengajukan klaim konversi SKS berdasarkan pengalaman kerja atau pendidikan sebelumnya, dengan melibatkan aktor Peserta, Ketua Program Studi, dan Biro Akademik.

Perbedaan mendasar antara kedua sistem ini memberikan ruang evaluasi yang komprehensif untuk penelitian. Perbandingan parameter antara SIM SKK dan RPL disajikan dalam Tabel 2.

Tabel 2. Matriks Karakteristik Studi Kasus

Parameter Perbandingan	SIM SKK	Sistem RPL	Landasan Pengujian
Sifat Alur Kerja	Dinamis	Rigid	Menguji ketahanan skrip otomatis terhadap variasi alur
Dominasi Data	Berbasif File (dokumen)	Berbasis Teks (naratif)	Membandingkan kecepatan eksekusi pada skenario <i>input-heavy</i>
Mekanisme Validasi	Verifikasi Bertingkat	Asesmen Akademik	Menilai akurasi deteksi bug pada logika bisnis yang berbeda
Interaksi Antarmuka	Berorientasi status menggunakan Poin	Berorientasi prosedur menggunakan SKS	Evaluasi <i>coverage</i> (cakupan pengujian) pada berbagai aktor

Dengan menempatkan kedua sistem ini sebagai objek studi, penelitian dapat memberikan landasan ilmiah yang kuat dalam menentukan metode pengujian yang paling optimal untuk pengujian manual atau otomatis berdasarkan karakteristik spesifik dari sebuah sistem informasi. Karakteristik kontras antara SIM SKK (dinamis) dan RPL (linear) yang telah ditetapkan kemudian menjadi basis dalam perancangan instrumen uji, di mana setiap kebutuhan fungsional ditransformasikan menjadi skenario dan *test case modular* agar dapat dieksekusi secara identik oleh manusia maupun mesin.

2.4. Perancangan Instrumen Uji (Test Scenario dan Test Case)

Tahap perancangan instrumen uji merupakan fase krusial untuk menjamin validitas perbandingan antara metode manual dan otomatis. Pada tahap ini, dilakukan transformasi kebutuhan fungsional sistem ke dalam unit pengujian yang terukur dan terstandarisasi.

Pengembangan Test Scenario: *Test scenario* disusun sebagai representasi dari alur bisnis utama pada kedua objek penelitian. Skenario ini berfungsi sebagai klasifikasi pengujian tingkat tinggi yang mencakup seluruh fitur. Skenario SIM SKK berfokus pada manajemen aktivitas mahasiswa, yang mencakup 6 skenario utama yaitu SKK-01 hingga SKK-06, sedangkan Sistem RPL memiliki cakupan yang lebih luas dan detail dengan total 13 skenario utama yaitu RPL-01 hingga RPL-13 yang diuraikan pada Tabel 3.

Tabel 3. Test Scenario SIM SKK dan Sistem RPL

SIM SKK		Sistem RPL	
Kode	Fitur	Kode	Fitur
SKK-01	Login	RPL-01	Login
SKK-02	Tambah Kegiatan	RPL-02	Data Diri
SKK-03	Lihat Kegiatan	RPL-03	Data Orang Tua
SKK-04	Edit Kegiatan	RPL-04	Data Pendidikan
SKK-05	Hapus Kegiatan	RPL-05	Dokumen Persyaratan
SKK-06	Download Bukti SKK	RPL-06	Riwayat Pendidikan

SIM SKK		Sistem RPL	
Kode	Fitur	Kode	Fitur
		RPL-07	Pelatihan
		RPL-08	Seminar
		RPL-09	Penghargaan
		RPL-10	Organisasi
		RPL-11	Riwayat Pekerjaan
		RPL-12	Master File
		RPL-13	Form Evaluasi

Transformasi ke Test Case Modular: Setiap *test scenario* yang telah diidentifikasi kemudian diturunkan menjadi *test case modular*. Struktur *test case* dirancang secara detail agar dapat dieksekusi secara identik oleh penguji manusia maupun oleh mesin (otomatis melalui Katalon Studio). Struktur rancangan *test case* dalam penelitian ini disajikan pada Tabel 4.

Tabel 4. Struktur Rancangan Test Case

Komponen	Keterangan
<i>Test Case ID</i>	Berisi kode unik yang merujuk pada test scenario untuk menjamin ketertelusuran kebutuhan.
<i>Pre-condition</i>	Menetapkan parameter awal sistem, seperti status otentikasi aktor atau ketersediaan data pada basis data, agar pengujian berjalan dalam kondisi yang terkontrol.
<i>Test Steps</i>	Urutan prosedur teknis, mulai dari pengisian field teks, pemilihan dropdown, hingga interaksi tombol.
<i>Test Data</i>	Input spesifik yang disiapkan untuk memicu respon sistem, termasuk file unggahan dalam format tertentu.
<i>Expected Result</i>	Standar output atau perilaku sistem yang dinyatakan benar sesuai dengan spesifikasi kebutuhan.
<i>Actual Result</i>	Ruang untuk mencatat respon sistem yang terjadi secara nyata saat eksekusi dilakukan. Komponen ini krusial untuk mendokumentasikan deviasi atau perilaku sistem yang tidak terduga.
<i>Status (Pass/Fail)</i>	Penentuan akhir efikasi fitur. Status dinyatakan Pass jika Actual Result sesuai dengan Expected Result, dan dinyatakan Fail jika terjadi ketidaksesuaian atau anomali.

2.5. Implementasi Pengujian

Eksekusi pengujian dilakukan pada level system testing di *staging environment* (lingkungan pementasan) untuk menghindari intervensi data eksternal, dengan spesifikasi sebagai berikut:

- Prosesor: Intel Core i7-10750 @ 2.60GHz
- Memori (RAM): 16 GB DDR4
- Sistem Operasi: Windows 10 Home 64-bit
- Koneksi Internet: dedicated 10 Mbps

Metode pengujian yang diterapkan adalah *Checklist-based Testing*, di mana dokumen *checklist* diturunkan langsung dari dekomposisi fitur pada test scenario yang telah divalidasi. Dokumen checklist ini bertindak sebagai matriks kendali guna memastikan seluruh fungsi kritikal pada SIM SKK dan RPL dievaluasi tanpa ada yang terlewat.

Pengujian Manual

Pengujian manual dieksekusi secara langsung oleh penguji dengan mensimulasikan aktivitas pengguna riil berdasarkan hak akses masing-masing aktor. *Checklist-based Testing* pada metode manual dilakukan melalui tahapan berikut:

- a. Persiapan Lembar Kerja: penguji membuka dokumen checklist yang berisi daftar fungsi spesifik yang harus divalidasi, misalnya "Validasi format file .pdf pada unggah sertifikat SIM SKK".
- b. Eksekusi Prosedural: penguji melakukan interaksi fisik dengan antarmuka web, melakukan input data secara manual, kemudian mengklik tombol aksi sesuai dengan urutan langkah kerja.
- c. Observasi Visual dan Logika: penguji mengamati respon visual sistem, seperti pesan kesalahan, perubahan status data, dan kebenaran navigasi halaman.
- d. Pencatatan Hasil: penguji membandingkan perilaku sistem dengan dokumen *checklist*. Jika sistem berperilaku sesuai ekspektasi, penguji menandai status *Pass* pada *checklist*, dan jika tidak sesuai maka menandai status *Fail* pada kolom *Actual Result*.

Pengujian Otomatis

Pengujian otomatis diimplementasikan menggunakan perangkat lunak Katalon Studio dengan memanfaatkan bahasa pemrograman Java. Untuk menjaga konsistensi parameter uji, pendekatan *Checklist-based Testing* pada metode otomatis ditranslasikan ke dalam arsitektur skrip pemrograman dengan mekanisme sebagai berikut:

- a. Penyusunan *Object Repository*: setiap elemen antarmuka seperti tombol, *text field*, dan *dropdown* pada SIM SKK dan RPL diidentifikasi menggunakan pencari lokasi seperti *XPath* atau *CSS Selector*, kemudian disimpan secara modular ke dalam *Object Repository* Katalon.
- b. Kodifikasi *Checklist* menjadi *Test Script*: setiap butir poin pada dokumen *checklist* diubah menjadi baris kode perintah. Sebagai contoh, poin checklist "Verifikasi input nama di RPL" ditranslasikan menjadi skrip otomatis seperti:

```
WebUI.setText(findTestObject('Object  
Repository/RPL/input>Nama'), 'Budi Sudarsono')
```

- c. Implementasi *Automated Assertion*: peran observasi manusia digantikan oleh fungsi *Assertion* (pernyataan program). Katalon Studio diprogram untuk memvalidasi respon sistem secara otomatis menggunakan perintah seperti:

```
WebUI.verifyElementPresent(findTestObject('Object  
Repository/SKK/alert_Sukses'), 5)
```

Jika objek atau respon yang dicari muncul dalam *timeout* (batas waktu) 5 detik, skrip akan memberikan status *Pass* ke dalam Log Viewer, dan otomatis memberikan status *Fail* serta menghentikan *test suite* jika terjadi kegagalan logika sistem.

Parameter Konsistensi Eksperimen

Agar data perbandingan kriteria waktu pengujian, biaya pengujian, cakupan pengujian, dan penemuan *bug* bersifat valid, pengujian manual dan otomatis dikondisikan pada parameter yang setara, seperti yang diuraikan pada Tabel 5.

Tabel 5. Konsistensi Operasional Checklist-based Testing

Aktivitas Pengujian	Implementasi Manual	Implementasi Otomatis (Katalon)	Target Validasi
Input Data	Pengetikan secara manual oleh penguji	Pemanggilan variabel melalui script	Kesamaan beban data
Navigasi Alur	Klik tautan atau tombol secara sekunsial	Eksekusi perintah <code>WebUI.click()</code>	Kesamaan alur logika
Verifikasi Output	Observasi layar pada monitor	Eksekusi fungsi <code>WebUI.verify()</code>	Kesamaan standar kelulusan pengujian

Format Lembar Kerja Checklist-based Testing

Dokumen ini dirancang modular agar dapat dieksekusi secara Manual oleh penguji melalui observasi visual dan secara Otomatis yang ditranslasikan menjadi fungsi *assertion* pada Katalon Studio.

Contoh *checklist* untuk SIM SKK pada fitur SKK-02 (Tambah Kegiatan Mahasiswa) diuraikan pada Tabel 6.

Tabel 6. Contoh Checklist SIM SKK

ID Checklist	Item Validasi	Metode Manual	Metode Otomatis	Hasil Yang Diharapkan
CL-SKK-02-01	Validasi input teks pada kolom isian wajib (nama kegiatan, penyelenggara, dan tingkat).	Mengetik teks alfanumerik pada seluruh kolom isian wajib.	<code>WebUI.setText()</code> menggunakan variabel data pengujian.	Sistem berhasil menyimpan teks tanpa memunculkan pesan <i>error</i> .
CL-SKK-02-02	Validasi batas ukuran file dokumen sertifikat yang diunggah.	Mengunggah file PDF dengan ukuran melebihi batas, misal lebih besar dari 5MB.	<code>WebUI.uploadFile()</code> dengan jalur file dokumen eksperimen.	Sistem menolak unggahan dan menampilkan pesan "file size exceeds limit".

Contoh *checklist* untuk Sistem RPL pada fitur RPL-02 (Pengisian Formulir Data Diri Calon Mahasiswa) diuraikan pada Tabel 7.

Tabel 7. Contoh Checklist Sistem RPL

ID Checklist	Item Validasi	Metode Manual	Metode Otomatis	Hasil Yang Diharapkan
CL-RPL-02-01	Validasi format nomor kontak ponsel pada kolom isian Nomor HP.	Memasukkan input karakter huruf (alfabet) pada kolom isian Nomor HP.	<code>WebUI.setText()</code> dengan nilai data uji string berisi huruf.	Sistem menolak input atau memunculkan peringatan "format nomor HP tidak valid".
CL-RPL-02-02	Validasi <i>dropdown cascading</i> untuk	Memilih Provinsi Jawa Barat, kemudian memeriksa	<code>WebUI.selectOptionByLabel()</code> diikuti fungsi	Pilihan Kabupaten/Kota otomatis

ID Checklist	Item Validasi	Metode Manual	Metode Otomatis	Hasil Yang Diharapkan
	sinkronisasi data wilayah.	opsi Kabupaten atau Kota.	WebUI.verifyOptionPresent().	berubah menyesuaikan Provinsi yang dipilih.

Hasil aktual dari eksekusi pengujian selanjutnya menjadi input performa yang akan diintegrasikan dengan bobot prioritas yang dihasilkan dari analisis AHP untuk menentukan skor global akhir.

2.6. Analisis Perbandingan Kriteria AHP

Metode AHP digunakan untuk menentukan bobot prioritas kriteria berdasarkan perspektif ahli.

Karakteristik Responden

Penilaian kriteria dilakukan melalui *expert judgement* yang melibatkan lima responden terpilih. Responden terdiri dari 4 orang praktisi *Quality Assurance* dengan pengalaman 1-3 tahun, dan dan 1 orang *Project Manager* yang memiliki pengalaman kerja 2 tahun dalam pengelolaan proyek perangkat lunak.

Penyusunan Kuesioner

Kuesioner disusun menggunakan matriks *pairwise comparison* (perbandingan berpasangan) berdasarkan skala fundamental *Saaty* (1-9). Jumlah kombinasi pertanyaan dihitung menggunakan Persamaan (1).

$$C_r^n = \frac{n!}{r!(n-r)!} \quad (1)$$

Dengan $n = 4$ kriteria dan $r = 2$ pilihan dalam satu kombinasi, maka diperoleh total 6 pertanyaan perbandingan berpasangan.

Perhitungan AHP

Data penilaian responden diolah melalui normalisasi matrik untuk menghasilkan bobot prioritas (w). Validitas hasil pembobotan diuji melalui *Consistency Ratio* (CR) untuk memastikan tidak ada bias logis dalam penilaian. Langkah-langkahnya meliputi:

- Menghitung *Consistency Index* (CI) menggunakan Persamaan (2).

$$CI = \frac{\lambda_{maks} - n}{n - 1} \quad (2)$$

Dimana, λ_{maks} adalah nilai *eigen* maksimum yang digunakan untuk mengukur apakah matriks perbandingan berpasangan (bobot prioritas) yang dibuat oleh responden sudah konsisten atau acak.

- Menghitung *Consistency Ratio* (CR) menggunakan persamaan (3).

$$CR = \frac{CI}{RI} \quad (3)$$

Data dinyatakan valid secara ilmiah jika nilai $CR < 0,1$. Sedangkan *RI* adalah *Random Index* yang sesuai dengan ordo matriks. Nilai *RI* dapat dilihat pada Tabel 8.

Tabel 8. Konsistensi Operasional Checklist-based Testing

n	RI
1	0
2	0
3	0,58
4	0,9
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49

Untuk ukuran matriks perbandingan berpasangan yang digunakan adalah 6, sehingga nilai *RI* adalah 1,24.

Penentuan Skor Global

Tahap akhir adalah mengintegrasikan bobot kriteria (*w*) yang dihasilkan dari AHP dengan data performa eksekusi (*p*) dari kedua studi kasus. Skor global (*S*) untuk setiap alternatif dihitung dengan menjumlahkan hasil kali bobot kriteria dengan nilai performa kriteria tersebut menggunakan Persamaan (4).

$$S_{global} = \sum_{i=1}^n w_i p_i \quad (4)$$

Apabila ditemukan perbedaan penilaian yang kontras antar pakar yang menyebabkan nilai CR melampaui ambang batas 0,1, maka dilakukan proses konsolidasi persepsi melalui iterasi kuesioner lanjutan. Hasil penilaian individu disinkronisasikan menggunakan metode *Geometric Mean* untuk memperoleh matriks keputusan kelompok yang representatif dan stabil secara matematis sebelum dilanjutkan ke tahap perhitungan skor global.

3. HASIL DAN PEMBAHASAN

3.1. Perancangan Instrumen dan Eksekusi Pengujian

Instrumen pengujian diturunkan secara terstruktur dari hasil wawancara bersama *product owner* masing-masing sistem untuk menjamin *Requirement Traceability*. Kebutuhan fungsional tersebut ditransformasikan menjadi *test scenario* tingkat tinggi, yang kemudian didekomposisi menjadi butir *test case* modular. Untuk menjaga objektivitas eksperimen, jumlah skenario dan prosedur *test case* yang dirancang diberlakukan secara identik pada metode manual maupun otomatis. Pemetaan instrumen uji tersebut dirangkum dalam Tabel 9.

Tabel 9. Contoh Checklist Sistem RPL

Objek Sistem	Kode Skenario Utama	Fokus Cakupan Fitur	Jumlah Test Case
SIM SKK	SKK-01 sampai SKK-06	Login, Tambah Kegiatan, Lihat, Edit, Hapus, & Download Bukti	108
Sistem RPL	RPL-01 sampai dengan RPL-13	Login, Data Diri, Orang Tua, Pendidikan, Dokumen, Pelatihan, Seminar, Penghargaan, Organisasi, Pekerjaan, Master File, & Evaluasi Diri	142

3.2. Analisis Hasil Eksperimen

Pengujian pada SIM SKK melibatkan 108 *test case* fungsional dengan fokus pada evaluasi alur kerja dinamis, sedangkan pengujian pada sistem RPL melibatkan 142 *test case* fungsional untuk mengevaluasi ketahanan terhadap alur kerja linear dan kaku. Data empiris yang dihasilkan dari eksekusi kedua sistem ini dievaluasi berdasarkan empat kriteria utama: Waktu Pengujian, Cakupan Pengujian, Biaya Pengujian, dan Penemuan *Bug*.

Hasil Perbandingan Pengujian Pada Sistem

Untuk memberikan gambaran yang komprehensif mengenai efisiensi dan efektivitas dari kedua metode pengujian, hasil eksperimen dievaluasi berdasarkan empat kriteria utama: Waktu Pengujian, Cakupan Pengujian, Biaya Pengujian menggunakan pendekatan *Time-Driven Activity-Based Costing* (TDABC), dan Penemuan *Bug* atau Defek. Rekapitulasi data empiris hasil pengujian pada Sistem Informasi Manajemen Satuan Kredit Kemahasiswaan (SIM SKK) dan Sistem Rekognisi Pembelajaran Lampau (RPL) disajikan secara terstruktur pada Tabel 10.

Tabel 10. Matriks Perbandingan Performa Pengujian Manual dan Otomatis

Objek Sistem	Kriteria Evaluasi Performa	Pengujian Manual	Pengujian Otomatis
SIM SKK (108 <i>test case</i>)	Waktu Pengujian	3.956 detik	1.870 detik
	Cakupan Pengujian	100% (108 TC)	100% (108 TC)
	Biaya Pengujian	Rp595.680,-	Rp1.494.920,-
	Penemuan <i>Bug</i>	29 Bug	33 Bug
Sistem RPL (142 <i>test case</i>)	Waktu Pengujian	4.203 detik	2.140 detik
	Cakupan Pengujian	97% (139 TC)	89% (127 TC)
	Biaya Pengujian	Rp724.880,-	Rp1.575.160,-
	Penemuan <i>Bug</i>	35 Bug	40 Bug

Berdasarkan data empiris pada "Tabel 10", dapat dilihat bahwa masing-masing kriteria memiliki satuan ukur dan rentang nilai nominal yang sangat beragam dan kontras. Kriteria waktu diukur dalam satuan detik, cakupan pengujian menggunakan skala persentase, biaya operasional dihitung dalam satuan mata uang Rupiah, sedangkan kemampuan penemuan bug menggunakan skala kuantitas jumlah kasus bug. Oleh karena itu, diperlukan sebuah proses transformasi data melalui tahap normalisasi untuk menyeragamkan skala rasio dari 0 sampai 1 menggunakan Persamaan (5) untuk Cakupan dan Penemuan Bug, dan Persamaan (6) untuk Waktu Pengujian dan Biaya Pengujian.

$$p_{ij} = \frac{x_{ij}}{x_{j \max}} \quad (5)$$

Dimana, p_{ij} adalah Nilai performa hasil normalisasi untuk alternatif i pada kriteria j , x_{ij} adalah nilai performa aktual alternatif i pada kriteria j , dan $x_{j \max}$ adalah nilai performa aktual tertinggi yang tercatat pada kriteria j di antara metode yang dibandingkan.

$$p_{ij} = \frac{x_{j \min}}{x_{ij}} \quad (6)$$

Dimana, $x_{j \min}$ adalah nilai performa aktual terendah yang tercatat pada kriteria j di antara metode yang dibandingkan.

Deskripsi Performa Kriteria Berdasarkan Karakteristik Sistem

Hasil eksperimen yang dirangkum dalam "Tabel 11" menunjukkan perbedaan performa yang signifikan antara kedua metode. Pada SIM SKK, pengujian otomatis mampu mempertahankan cakupan uji 100% dengan efisiensi waktu mencapai 52,7% dibandingkan manual. Sebaliknya, pada Sistem RPL, karakteristik alur yang rigid menyebabkan munculnya potensi kerapuhan skrip terhadap alur kerja yang mengakibatkan penurunan cakupan uji otomatis menjadi 89%, sementara manual tetap unggul di 97% karena fleksibilitas pengujian manusia.

Komponen Biaya Berbasis TDABC

Kriteria biaya dalam penelitian ini dihitung menggunakan pendekatan *Time-Driven Activity-Based Costing* (TDABC). Biaya yang tercatat pada "Tabel 10" tidak hanya mencakup durasi aktivitas pengujian (tarif SDM per menit), namun juga mengintegrasikan investasi pada perangkat lunak yang diperlukan. Hal ini menjelaskan mengapa biaya pengujian otomatis (sekitar Rp1,5 juta) jauh lebih tinggi dibandingkan manual (sekitar Rp600-700 ribu), karena adanya biaya lisensi atau pemeliharaan alat bantu seperti Katalon Studio di samping kapasitas biaya sumber daya manusia.

Konversi Data Faktual Menjadi Nilai Performa (Normalisasi)

Mengingat setiap kriteria memiliki satuan ukur yang berbeda, yaitu detik, persentase, rupiah, dan jumlah bug, maka dilakukan transformasi data melalui tahap normalisasi untuk menyeragamkan skala rasio menjadi 0 hingga 1. Nilai 1 merepresentasikan performa terbaik pada kriteria tersebut. Proses konversi ini menggunakan dua pendekatan matematis:

- a. Kriteria "semakin besar maka semakin baik" untuk kriteria Cakupan dan Penemuan Bug): menggunakan Persamaan (5), di mana nilai aktual dibagi dengan nilai tertinggi yang tercatat di antara kedua metode.
- b. Kriteria "semakin kecil semakin baik" untuk Waktu dan Biaya: menggunakan Persamaan (6), di mana nilai terendah yang tercatat dibagi dengan nilai aktual.

Validitas Penilaian dan Representativitas

Meskipun hasil eksperimen bersifat faktual, interpretasi kepentingan kriteria didasarkan pada penilaian 5 responden pakar. Jumlah responden ini dipandang memadai untuk mewakili perspektif industri karena kualifikasi spesifik mereka (QA dan *Project Manager*) dan validitas ilmiahnya telah diuji melalui *Consistency Ratio* (CR). Penelitian ini melewati tiga kali iterasi kuesioner hingga mencapai nilai $CR < 0,1$ pada iterasi terakhir, yang membuktikan bahwa data penilaian tersebut konsisten secara logis dan tidak bias sebelum diintegrasikan menjadi Skor Global.

3.3. Pembobotan Kriteria Menggunakan AHP

Proses pengumpulan data melewati tiga iterasi kuesioner perbandingan berpasangan (*pairwise comparison*) yang tidak hanya mengandalkan satu kali penilaian. Dalam penelitian ini, proses pengumpulan data bahkan melewati tiga iterasi kuesioner karena evaluasi pada iterasi pertama dan kedua belum memenuhi ambang batas konsistensi ilmiah. Metode sinkronisasi *Geometric Mean* digunakan untuk menyatukan berbagai perspektif pakar yaitu 4 orang QA dan 1 orang Project Manager menjadi satu keputusan kelompok. Metode ini berfungsi sebagai instrumen sinkronisasi yang mampu mereduksi dampak dari nilai-nilai ekstrem dari responden tertentu, sehingga menghasilkan matriks keputusan grup yang stabil. Validasi Akhir dilakukan dengan cara proses sinkronisasi yang dinyatakan berhasil secara ilmiah apabila hasil penggabungan tersebut menghasilkan matriks dengan nilai Consistency Ratio (CR) $< 0,1$. Pada penelitian ini, setelah sinkronisasi iterasi ketiga, diperoleh matriks yang konsisten dengan nilai Eigen Maksimum sebesar 4,08. Hasil pembobotan prioritas kriteria dari perhitungan akhir AHP disajikan pada Tabel 11.

Tabel 11. Nilai Bobot Kriteria Akhir

Kriteria Pengujian	Nilai Bobot Prioritas (w_i)	Peringkat Kepentingan Alternatif
Penemuan Bug (C1)	0,4334	Peringkat 1 (prioritas utama)
Cakupan Pengujian (C2)	0,4233	Peringkat 2
Waktu Pengujian (C3)	0,0929	Peringkat 3
Biaya Pengujian (C4)	0,0504	Peringkat 4 (prioritas terendah)

Data pada "Tabel 11", dipaparkan bahwa bagi para pakar di industri, aspek kualitas akhir perangkat lunak yang diwakili oleh kemampuan Penemuan Bug (43,34%) dan Cakupan Pengujian (42,33%) merupakan faktor paling krusial, melampaui efisiensi teknis berupa waktu maupun biaya pengujian.

3.4. Perhitungan Skor Global dan Penentuan Keputusan

Setiap kriteria memiliki satuan ukur yang berbeda (detik, persentase, rupiah, dan jumlah bug), sehingga dilakukan normalisasi nilai performa (p_i) ke dalam skala rasio 0 hingga 1 menggunakan persamaan optimasi. Nilai 1 merepresentasikan performa terbaik pada kriteria tersebut. Skor Global akhir diperoleh dengan mengalikan nilai normalisasi dengan bobot prioritas AHP ($w_i \times p_i$).

Evaluasi Skor Global Pada SIM SKK

Pada sistem SIM SKK, proses normalisasi data performa memberikan hasil yang kontras pada beberapa kriteria. Pengujian otomatis mendominasi secara mutlak dengan meraih skor penuh (nilai 1,000) pada kriteria waktu eksekusi dan kriteria penemuan bug. Sementara itu, karena kedua metode sama-sama mencapai cakupan pengujian sempurna sebesar 100%, kriteria coverage untuk kedua alternatif mendapatkan nilai normalisasi maksimal (nilai 1,000). Matriks perhitungan skor global untuk studi kasus SIM SKK dijabarkan pada Tabel 12.

Tabel 12. Hasil Perhitungan Skor Global Pada SIM SKK

Kriteria Pengujian	Bobot AHP (w)	Nilai Normalisasi Manual (p_m)	Skor Akhir Manual	Nilai Normalisasi Otomatis (p_o)	Skor Akhir Otomatis
Waktu Pengujian (C1)	0,0929	0,0473	0,0439	1,000	0,0929
Cakupan Pengujian (C2)	0,4233	1,000	0,4233	1,000	0,4233
Biaya Pengujian (C3)	0,0504	1,000	0,0504	0,398	0,0201
Penemuan Bug (C4)	0,4334	0,879	0,3810	1,000	0,4334
Total Skor Global	1,0000	-	0,8986	-	0,9697

Berdasarkan akumulasi perhitungan pada "Tabel 12", diperoleh nilai total skor akhir sebagai berikut:

- a. Skor Global Pengujian Manual SIM SKK = 0,8986
- b. Skor Global Pengujian Otomatis SIM SKK = 0,9697

Hasil ini menunjukkan bahwa pada karakteristik sistem informasi yang bersifat dinamis seperti SIM SKK, pengujian otomatis memberikan keunggulan yang sangat signifikan dengan selisih skor sebesar 0,0711. Keunggulan mutlak ini terjadi karena skrip otomatisasi mampu mengeksploitasi efisiensi waktu eksekusi secara penuh dan mendeteksi lebih banyak bug, tanpa mengalami kendala penurunan cakupan pengujian akibat hambatan *blocker* alur sistem.

Evaluasi Skor Global Pada Sistem RPL

Pada sistem RPL, pengujian manual berhasil meraih skor penuh (nilai 1,000) pada kriteria cakupan pengujian dan biaya pengujian. Sebaliknya, pengujian otomatis mendominasi kriteria waktu pengujian dan penemuan bug. Matriks perhitungan skor global untuk studi kasus RPL dijabarkan pada Tabel 13.

Tabel 13. Hasil Perhitungan Skor Global Pada Sistem RPL

Kriteria Pengujian	Bobot AHP (w)	Nilai Normalisasi Manual (p_m)	Skor Akhir Manual	Nilai Normalisasi Otomatis (p_o)	Skor Akhir Otomatis
Waktu Pengujian (C1)	0,0929	0,509	0,0473	1,000	0,0929
Cakupan Pengujian (C2)	0,4233	1,000	0,4233	1,918	0,3884
Biaya Pengujian (C3)	0,0504	1,000	0,0504	0,460	0,0232
Penemuan Bug (C4)	0,4334	0,875	0,3792	1,000	0,4334
Total Skor Global	1,0000	-	0,9002	-	0,9379

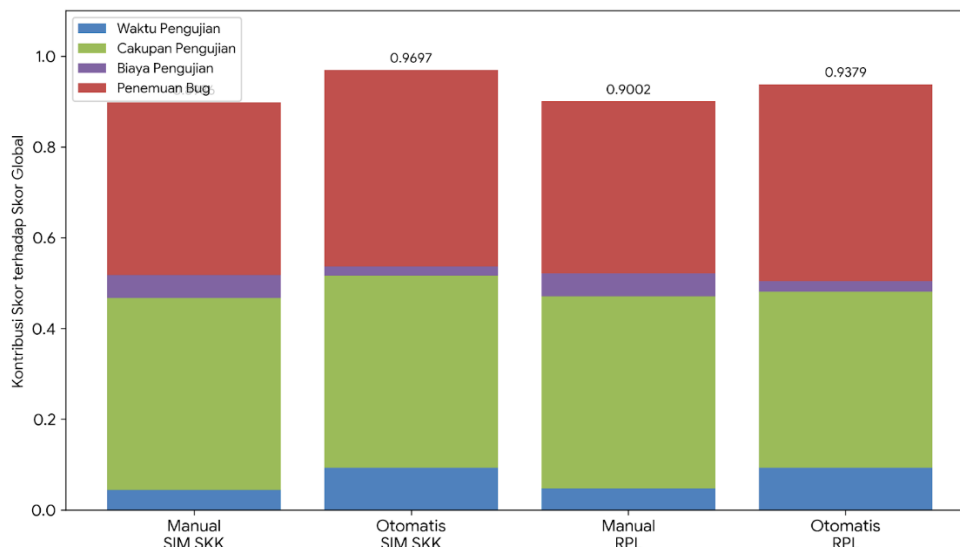
Berdasarkan akumulasi perhitungan pada "Tabel 13", diperoleh nilai total skor akhir sebagai berikut:

- a. Skor Global Pengujian Manual SIM SKK = 0,9002
- b. Skor Global Pengujian Otomatis SIM SKK = 0,9379

Meskipun pengujian otomatis mengalami penurunan performa pada kriteria cakupan pengujian akibat struktur alur proses RPL yang kaku dan linear, skor global pengujian otomatis menunjukkan yang paling optimal dengan bobot kepentingan yang jauh lebih tinggi sebesar 43,34% dalam model keputusan AHP dibandingkan kriteria lainnya. Hal ini membuktikan transisi dari pengujian manual ke pengujian otomatis menggunakan alat bantu seperti Katalon Studio memberikan peningkatan kualitas dan efisiensi pengujian perangkat lunak, baik karakteristik sistem yang bersifat dinamis maupun linear.

Visualisasi skor global perbandingan antara pengujian manual dan otomatis pada SIM SKK dan Sistem RPL dapat dilihat pada Gambar 2.

Perbandingan Efektivitas Pengujian Manual dan Otomatis Menggunakan System Testing dengan Pendekatan AHP



Gambar 2. Perbandingan Skor Global Pengujian Manual dan Otomatis Pada SIM SKK dan Sistem RPL

4. KESIMPULAN

Kesimpulan penelitian ini adalah sebagai berikut:

- Objektivitas Pengambilan Keputusan: penerapan metode *Analytic Hierarchy Process* (AHP) mampu menyediakan landasan kuantitatif yang objektif untuk memitigasi subjektivitas dalam pemilihan strategi pengujian. Kerangka kerja ini memungkinkan integrasi berbagai variabel performa ke dalam satu model keputusan yang terukur
- Prioritas Kualitas di Industri: temuan utama menunjukkan di tingkat praktisi aspek kualitas yaitu penemuan bug dan cakupan pengujian memiliki urgensi yang jauh lebih tinggi dibandingkan efisiensi teknis seperti waktu dan biaya. Hal ini menegaskan bahwa strategi pengujian yang optimal adalah yang mampu menjamin keamanan sistem sebelum dirilis
- Dampak Karakteristik Arsitektur Sistem: efektivitas pengujian otomatis sangat bergantung pada karakteristik alur kerja sistem. Sistem dengan alur kerja dinamis memberikan ruang bagi otomatisasi untuk mencapai performa maksimal. Sebaliknya, sistem dengan alur linear yang kaku memicu tantangan berupa kerapuhan skrip yang dapat menghambat cakupan uji otomatis, sehingga pada kondisi ini fleksibilitas pengujian manual tetap memiliki peran krusial.
- Keunggulan Strategi Otomatisasi: Secara keseluruhan, penggunaan alat bantu otomatisasi tetap merupakan pilihan paling optimal untuk meningkatkan kualitas perangkat lunak secara sistematis. Dominasi otomatisasi pada kriteria penemuan bug dan kecepatan eksekusi menjadikannya solusi yang paling efektif dalam mendukung SDLC yang efisien.

DAFTAR RUJUKAN

- Abdulwareth, A. J., & Al-shargabi, A. A. (2021). Towards a Multi-Criteria Framework for Selecting Software Testing Tools. *IEEE Access*, *XX*.
<https://doi.org/10.1109/ACCESS.2021.3128071>
- Ahmad, Y., & Hinchey, M. (2023). Optimizing Manual Testing Processes in Software Development: A Manual Tester 's Perspective Submitted to the University of Limerick , August 2023 in Partial Science in Software Engineering Supervised By: *University of Limerick Ollscoil Luimnigh, August*.
- Ardic, B., Brandt, C., Khatami, A., Swillus, M., & Zaidman, A. (2025). The Journal of Systems & Software The qualitative factor in software testing : A systematic mapping study of. *The Journal of Systems & Software*, *22*(October 2024), 112447.
<https://doi.org/10.1016/j.jss.2025.112447>
- Dewi, I. A., Fitria, L., Desrianty, A., & Afifah, A. U. (2020). The Development of Company Profile Website for CV . Rian using Waterfall Model SDLC. *REKA ELKOMIKA: Jurnal Pengabdian Kepada Masyarakat*, *1*(2), 75–85.
- Honest, N. (2019). Role of Testing in Software Development Life Cycle. *International Journal of Computer Sciences and Engineering*, *7*(5), 886–889.
- Hossain, M. I. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management. *International Journal for Multidisciplinary Research (IJFMR)*, *5*(5), 1–36.
- Huda, B., Hananto, A. L., Studi, P., Informasi, S., Komputer, F. I., Buana, U., Karawang, P., Framework, R., & Life, S. T. (2023). Penerapan Software Testing Life Cycle Pada Pengujian Otomatisasi Platform Dzikra. *CSRID Journal*, *15*(1), 1–11.
- Kumari, B., Chauhan, N., & Vedpal. (2018). A COMPARISON BETWEEN MANUAL TESTING AND AUTOMATED TESTING. *Journal of Emerging Technologies and Innovative Research*, *5*(12), 323–332.
- Min, J. L., Istiqomah, A., & Rahmani, A. (2020). EVALUASI PENGGUNAAN MANUAL DAN AUTOMATED SOFTWARE TESTING PADA PELAKSANAAN END-TO-END TESTING. *JTT (Jurnal Teknologi Terapan)*, *6*(1).
- Putri, Y. F., Irianto, A. B. P., & Sharma, S. (2025). Comparison of Automatic and Manual Regression Testing on Mobile Application Health Technology with Black Box Testing Method. *Indonesian Journal of Information Systems (IJIS)*, *7*(2), 218–230.
- Riski, M., & Renanti, M. D. (2024). ANALISIS PERBANDINGAN MANUAL TESTING DAN AUTOMATION TESTING PADA SISITEM INFORMASI HUMAN RESOURCE. *KURAWAL*

Jurnal Teknologi, Informasi Dan Industri, 7(October), 1–11.

- Rochimah, S., Laili, U., & Balqis, S. (2023). Software Quality Measurement for Functional Suitability , Performance Efficiency , and Reliability Characteristics Using Analytical Hierarchy Process. *International Journal On Informatics Visualization*, 7(December), 2421–2426.
- Thant, K. S., & Tin, H. H. K. (2023). The Impact of Manual and Automatic Testing on Software Testing Efficiency and Effectiveness. *Indian Journal of Science and Research*, 3(3), 88–93.
- Yusuf, A. A. (2022). University of Ibadan Qualitative Comparative Analysis of Software Integration Testing Techniques. *Journal of Science and Logics in ICT Research*, 7(2), 67–82.
- Zafar, M. N., & Enoiu, E. (2022). Evaluating System-Level Test Generation for Industrial Software : A Comparison between Manual , Combinatorial and Model-Based Testing. In *In IEEE/ACM3rd form concerning both structural and system-level metrics such as International Conference on Automation of Software Test (AST '22)* (Vol. 1, Issue 1). Association for Computing Machinery. <https://doi.org/10.1145/3524481.3527235>
- Zarrad, A., Bahsoon, R., & Manimaran, P. (2024). Optimizing regression testing with AHP - TOPSIS metric system for effective technical debt evaluation. *Automated Software Engineering*, 31(2), 1–28. <https://doi.org/10.1007/s10515-024-00458-5>