

Analisis Efektivitas Regression Testing Menggunakan Metrik Kuantitatif pada Aplikasi Digital Banking: Studi Kasus BLU by BCA Digital

Thomy Kurniawan¹, Dany Yudha Krisna², Debi Irawan³, Gita Cahyani Lestari⁴, Imam Maliki⁵

Sistem Informasi, Fakultas Ilmu Komputer, Universitas Indonesia Membangun (INABA)

E-mail: thomy.kurniawan@student.inaba.ac.id¹, dani.yudha@inaba.ac.id²

Article History:

Received: 28 Oktober 2025

Revised: 14 November 2025

Accepted: 16 November 2025

Keywords:

Regression Testing, Quantitative Metrics, Digital Banking, Quality Assurance, Test Automation.

Abstract: This study analyzes the effectiveness of regression testing using quantitative metrics in the BLU by BCA Digital digital banking application. The objective is to evaluate the effectiveness of the regression testing strategy and identify areas for improvement within the context of a complex financial application. The research employed an explanatory quantitative approach with a single case study design. Primary data was collected prospectively from testing artifacts and the CI/CD pipeline over three release cycles, encompassing 3,301 test scenarios. Data analysis utilized descriptive and inferential statistical techniques, focusing on quantitative metrics such as Defect Detection Percentage (DDP), Test Case Effectiveness (TCE), Requirements Coverage, and Test Execution Efficiency. The results indicate that regression testing was highly effective, achieving 100% completion with a functional failure rate of only 0.06%. A hybrid strategy combining automation (32.87% coverage) and manual testing proved optimal; automation provided high efficiency and stability, while manual testing played a crucial role in detecting complex defects. However, critical areas were identified, including performance disparities between application modules and a high automation skip rate (18.06%). In conclusion, the hybrid approach effectively ensures release quality, but strategic expansion of automation coverage and improved script maintainability are required. This research contributes to the development of a quantitative evaluation framework for quality assurance in the Indonesian digital banking industry.

Kata Kunci:

Pengujian Regresi, Metrik Kuantitatif, Bank Digital, Jaminan Kualitas, Pengujian Otomasi.

Abstrak: Penelitian ini menganalisis efektivitas regression testing menggunakan metrik kuantitatif pada aplikasi digital banking BLU by BCA Digital. Tujuannya adalah untuk mengevaluasi efektivitas strategi pengujian regresi dan mengidentifikasi area perbaikan dalam konteks aplikasi finansial yang kompleks. Metode yang digunakan adalah pendekatan kuantitatif eksplanatori dengan desain studi kasus tunggal. Data primer dikumpulkan secara prospektif dari artefak pengujian dan pipeline CI/CD selama tiga siklus rilis, mencakup 3.301 skenario pengujian. Analisis data menggunakan teknik statistik deskriptif dan inferensial, dengan fokus pada metrik kuantitatif seperti Defect Detection Percentage (DDP), Test Case Effectiveness (TCE), Requirements Coverage, dan Test

Execution Efficiency. Hasil penelitian menunjukkan bahwa regression testing berjalan sangat efektif dengan penyelesaian 100% dan tingkat kegagalan fungsional hanya 0,06%. Strategi hybrid yang memadukan automation (32,87% cakupan) dan manual testing terbukti optimal; automation memberikan efisiensi dan stabilitas tinggi, sementara manual testing berperan krusial dalam mendeteksi defect kompleks. Namun, ditemukan disparitas kinerja antar modul dan tingginya skip rate automation (18,06%) sebagai area kritis. Kesimpulannya, pendekatan hybrid efektif menjamin kualitas rilis, namun diperlukan ekspansi cakupan otomatisasi dan perbaikan maintainability script. Penelitian ini berkontribusi pada pengembangan framework evaluasi kuantitatif untuk quality assurance di industri digital banking Indonesia.

PENDAHULUAN

Transformasi digital dalam industri perbankan telah mengubah paradigma layanan keuangan secara fundamental di seluruh dunia. Menurut *Challa (2025)*, revolusi teknologi finansial telah mendorong bank-bank tradisional untuk mengadopsi platform cloud-native yang scalable guna mempertahankan daya saing di era digital. Fenomena ini tidak hanya mengubah cara bank beroperasi, tetapi juga meningkatkan ekspektasi konsumen terhadap kecepatan, keamanan, dan reliabilitas layanan perbankan digital. *Palanisamy (2025)* menekankan bahwa quality engineering telah menjadi komponen krusial dalam memastikan skalabilitas dan keandalan software finansial di lingkungan cloud-native. Dalam konteks ini, aplikasi digital banking tidak lagi sekadar kanal tambahan, melainkan menjadi tulang punggung operasional perbankan modern yang melayani jutaan transaksi setiap hari.

Kompleksitas aplikasi digital banking modern menuntut pendekatan quality assurance yang lebih sophisticated dan terukur. *Patel (2025)* mengidentifikasi bahwa quality engineering dalam layanan finansial modern memerlukan framework yang mengintegrasikan aspek keamanan, compliance, dan resilience secara holistik. Setiap update aplikasi menghadirkan risiko potensial terhadap fungsi-fungsi existing yang telah stabil, sehingga memerlukan mekanisme verifikasi yang komprehensif. *Algubelli (2025)* menyatakan bahwa regression testing merupakan metodologi kritis dalam safeguarding sistem finansial melalui verifikasi metodis terhadap setiap perubahan kode. *Hilman dan Mantiri (2025)* lebih lanjut menjelaskan bahwa automation dan teknik selection dalam regression testing telah menjadi fokus utama penelitian empiris untuk meningkatkan efisiensi proses pengujian tanpa mengorbankan coverage.

Industri fintech Indonesia mengalami pertumbuhan eksponensial yang menghadirkan tantangan unik dalam konteks quality assurance. *Billy Chandra (2022)* menganalisis bahwa keputusan penggunaan aplikasi BLU by BCA Digital dipengaruhi oleh berbagai faktor kualitas layanan yang menjadi concern utama pengguna digital banking di Indonesia. *Sodhi et al. (2025)* mengungkapkan bahwa performa aplikasi mobile banking yang security-enabled memerlukan analisis mendalam untuk memastikan keseimbangan antara keamanan dan user experience. Dalam ekosistem digital banking yang kompetitif, kepuasan pengguna menjadi diferensiator utama yang secara langsung berkorelasi dengan kualitas teknis aplikasi. *Dany Yudha Krisna et al. (2023)* menegaskan bahwa penerapan metode end user computing satisfaction efektif untuk menilai kualitas kepuasan pengguna aplikasi digital di Jakarta, mengindikasikan pentingnya measurement kuantitatif dalam evaluasi aplikasi.

Metodologi testing dalam pengembangan aplikasi mobile telah mengalami evolusi signifikan dari pendekatan manual ke automation testing. *Telaga et al. (2022)* membuktikan bahwa pengembangan automatic testing script mampu meningkatkan efisiensi testing untuk aplikasi mobile secara substansial. *Khaerunnisa et al. (2023)* melakukan comparative study terhadap Robot Framework dan Cucumber sebagai BDD automated testing tools, mengidentifikasi kelebihan dan keterbatasan masing-masing framework. *Srinivas dan Goel (2025)* menyajikan panduan komprehensif dalam mendesain dan mengimplementasikan robust test automation frameworks menggunakan Cucumber-BDD dan Java. *Jasmin dan Putri (2023)* mengkomparasikan metode automasi dan hybrid pada pengujian aplikasi mobile WebRTC menggunakan Appium, menunjukkan bahwa hybrid approach memberikan keseimbangan optimal antara coverage dan efisiensi.

Regression testing sebagai metodologi quality assurance memiliki peran fundamental dalam siklus pengembangan software yang iteratif. *Luginawati dan Wahyu (2023)* mengimplementasikan regression testing pada aplikasi pembelajaran matematika berbasis Android, memvalidasi efektivitas metode ini dalam memastikan stabilitas fungsi aplikasi setelah modifikasi. *Leung dan White (2021)* memberikan insights mendalam tentang regression testing yang menjadi foundation bagi penelitian-penelitian selanjutnya dalam domain ini. *Hussain et al. (2022)* mengevaluasi efektivitas regression testing dalam research dan practice, mengidentifikasi gap antara teori dan implementasi praktis. *Abdalla et al. (2022)* mengembangkan guidelines untuk mengombinasikan berbagai regression testing approaches, mengakui bahwa single approach seringkali tidak mencukupi untuk kompleksitas aplikasi modern.

Pengukuran efektivitas testing melalui metrik kuantitatif menjadi kebutuhan esensial dalam continuous delivery ecosystem. *Begum et al. (2025)* mengidentifikasi benefits dan challenges dari test automation software, menekankan pentingnya measurement objektif untuk mengevaluasi return on investment. *Rathaur et al. (2020)* menggunakan multiple linear regression untuk prediksi software defect density, mendemonstrasikan aplikasi statistical methods dalam quality assurance. *Cherkashin et al. (2021)* mengevaluasi efektivitas software testing process berdasarkan calculated metrics, mengembangkan framework matematis untuk assessment. *Salagundi dan Savitha (2024)* mengimplementasikan automated reporting dan statistical analysis hasil test case dalam continuous integration melalui custom dashboard approach, menunjukkan tren penggunaan data-driven decision making dalam quality engineering.

Pengembangan framework test automation yang robust menghadapi berbagai tantangan teknis dan organizational. *Dubey (2025)* melakukan comparative analysis terhadap tools dan frameworks untuk scalable software testing, mengidentifikasi kriteria selection yang tepat. *Neelapu (2025)* mengeksplorasi hybrid testing frameworks dengan fokus pada benefits dan challenges dalam automation, menyimpulkan bahwa kombinasi strategi testing memberikan flexibility optimal. *Mensah et al. (2023)* mengembangkan hybrid software testing model untuk meningkatkan software quality assurance, mengintegrasikan strengths dari berbagai metodologi. *Wakili et al. (2024)* menganalisis quality assurance practices dalam agile methodology, mengungkapkan bahwa regression testing harus adaptive terhadap sprint cycles yang rapid. *Sinaga dan Sitanggang (2023)* membandingkan performa Katalon Studio dan Appium pada aplikasi Investree, memberikan empirical evidence tentang trade-offs dalam tool selection.

Konteks quality assurance dalam aplikasi mobile memerlukan pertimbangan khusus terkait device fragmentation dan user behavior patterns. *Laily dan Triase (2023)* mengimplementasikan quality assurance dalam pengembangan aplikasi Ourticle berbasis Android, menghadapi challenges spesifik platform mobile. *Cui (2025)* menginvestigasi effects

dari mobile dan server-side automated testing terhadap DevOps performance efficiency, menemukan korelasi positif antara automation coverage dan deployment frequency. *Kumar (2021)* mengoptimasi cloud infrastructure untuk high-availability fintech services, menekankan bahwa testing strategy harus aligned dengan deployment architecture. *Bandi Sudakara (2025)* mengoptimalkan CI/CD pipeline di regulated industries dengan QA insights, mengakui bahwa compliance requirements menambah kompleksitas dalam testing workflow.

Meskipun berbagai studi telah mengeksplorasi regression testing dan test automation, terdapat research gap signifikan terkait pengukuran efektivitas menggunakan metrik kuantitatif dalam konteks spesifik digital banking Indonesia. *Richardson (2023)* mendiskusikan microservices patterns dengan distributed regression testing, namun aplikasi praktisnya dalam production environment masih terbatas. *Dubey (2025)* mereimajinasi test automation di AI era, namun framework evaluation yang systematic untuk financial applications masih langka. *Sharma dan Chande (2023)* mengoptimasi test case prioritization menggunakan machine learning algorithms, tetapi validasi empiris pada real-world banking applications belum memadai. *Silva dan Siriwardana (2023)* melakukan comparative study terhadap QA models across organizations, namun tidak fokus pada regression testing metrics di sektor finansial.

Penelitian ini bertujuan menganalisis efektivitas regression testing menggunakan metrik kuantitatif pada aplikasi BLU by BCA Digital, mengisi gap dalam literatur terkait measurement objektif quality assurance di digital banking Indonesia. Dengan mengintegrasikan perspektif dari *Ammann dan Offutt (2022)* tentang introduction to software testing, *Fenton dan Bieman (2022)* tentang software metrics rigorous approach, serta ISO/IEC 25010:2023 tentang product quality model, penelitian ini mengembangkan framework evaluasi komprehensif. Kontribusi teoretis penelitian meliputi pengembangan metrik efektivitas regression testing yang kontekstual untuk aplikasi finansial, sementara manfaat praktis mencakup rekomendasi actionable untuk optimasi testing strategy. *Alviana Putri et al. (2025)* menunjukkan pentingnya menganalisis pengaruh service quality terhadap customer satisfaction menggunakan Technology Acceptance Model, yang relevan sebagai outcome indicator dari effective regression testing. *Gita Cahyani Lestari (2024)* mengeksplorasi pengaruh kualitas sistem aplikasi DANA sebagai alat pembayaran, memberikan benchmark komparatif untuk evaluasi aplikasi digital banking. *Palupi Permata Rahmi et al. (2023)* menganalisis pengaruh kualitas layanan mobile banking terhadap kepuasan nasabah Bank BCA, mengkonfirmasi bahwa quality assurance yang efektif berkontribusi pada customer loyalty. *Kartika Bella Pertiwi et al. (2021)* meneliti kualitas pelayanan terhadap kepuasan dan loyalitas nasabah, memperkuat argumen bahwa investasi dalam quality engineering memberikan returns tangible dalam bentuk customer retention.

LANDASAN TEORI

1. Fundamental Quality Assurance dalam Aplikasi Digital Banking

Quality Assurance (QA) atau Jaminan Kualitas dalam rekayasa perangkat lunak merupakan suatu pendekatan sistematis dan terstruktur yang bertujuan untuk memastikan bahwa produk perangkat lunak memenuhi spesifikasi yang telah ditetapkan serta memuaskan kebutuhan pengguna. Berbeda dengan Quality Control (QC) yang berfokus pada identifikasi cacat pada produk akhir, QA bersifat prospektif dan mencakup seluruh siklus pengembangan perangkat lunak untuk mencegah terjadinya cacat sejak awal (*Escher, Neale, & Leusch, 2021*). Dalam konteks industri perbankan digital yang sangat dinamis dan diatur oleh ketentuan ketat, peran QA telah berevolusi dari sekadar fase verifikasi menjadi tulang punggung budaya pengembangan yang berkelanjutan. *Patel (2025)* menekankan bahwa quality engineering dalam layanan finansial

modern memerlukan kerangka kerja yang mengintegrasikan aspek keamanan, kepatuhan (compliance), dan ketahanan (resilience) secara holistik. Pergeseran paradigma ini didorong oleh transformasi digital yang mengubah layanan keuangan dari model tradisional menjadi platform cloud-native yang scalable, di mana keandalan, keamanan, dan kinerja aplikasi menjadi penentu utama daya saing dan kepercayaan nasabah (*Challa, 2025*).

Dalam ekosistem aplikasi digital banking, cakupan Quality Assurance meluas melampaui fungsionalitas dasar untuk mencakup dimensi-dimensi kritis yang secara langsung mempengaruhi pengalaman pengguna dan integritas sistem. Berdasarkan *ISO/IEC 25010:2023* tentang model kualitas produk, terdapat delapan karakteristik kualitas utama yang relevan untuk dievaluasi, yaitu: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, dan portability. Untuk aplikasi perbankan, dimensi keamanan (security) dan keandalan (reliability) menempati prioritas tertinggi mengingat sensitivitas data finansial dan transaksi yang ditangani. *Sodhi et al. (2025)* dalam analisisnya terhadap aplikasi mobile banking yang security-enabled mengungkapkan bahwa performa aplikasi memerlukan keseimbangan yang cermat antara implementasi kontrol keamanan yang ketat dan pengalaman pengguna yang mulus. Selain itu, *Billy Chandra (2022)* dalam penelitian mengenai keputusan penggunaan aplikasi BLU by BCA Digital menemukan bahwa berbagai faktor kualitas layanan, seperti keandalan, kecepatan respons, dan kemudahan penggunaan, merupakan concern utama pengguna yang secara signifikan mempengaruhi adopsi dan retensi. Temuan ini sejalan dengan penelitian *Palupi Permata Rahmi et al. (2023)* yang membuktikan adanya pengaruh positif kualitas layanan mobile banking terhadap kepuasan nasabah Bank BCA, serta penelitian *Kartika Bella Pertiwi et al. (2021)* yang mengonfirmasi hubungan antara kualitas pelayanan dengan kepuasan dan loyalitas nasabah.

Penerapan Quality Assurance pada aplikasi perbankan digital menghadapi tantangan unik yang membedakannya dari domain perangkat lunak lainnya. Kompleksitas arsitektur sistem yang sering kali mengadopsi pola microservices, integrasi dengan banyak pihak ketiga (payment gateway, credit scoring, dll.), dan tuntutan compliance terhadap regulasi keuangan seperti PBI (Peraturan Bank Indonesia) dan OJK (Otoritas Jasa Keuangan) menuntut pendekatan QA yang lebih sophisticated. *Kumar (2021)* menggarisbawahi bahwa strategi pengujian harus selaras dengan arsitektur deployment, terutama ketika mengoptimalkan infrastruktur cloud untuk layanan fintech yang beravailability tinggi. *Bandi Sudakara (2025)* menambahkan bahwa mengoptimalkan pipeline CI/CD di industri yang diatur (regulated industries) memerlukan integrasi insights QA yang mendalam untuk memastikan bahwa setiap release tidak hanya cepat tetapi juga mematuhi semua persyaratan compliance yang menambah kompleksitas dalam workflow pengujian. Tantangan organisasional juga muncul seiring dengan adopsi metodologi agile yang menuntut siklus pengujian yang adaptif terhadap sprint cycles yang rapid (*Wakili et al., 2024*). Oleh karena itu, praktik QA modern dalam digital banking tidak lagi dapat mengandalkan pendekatan manual tradisional, tetapi harus beralih ke model hybrid dan terotomasi yang diintegrasikan secara mulus ke dalam pipeline DevOps untuk memungkinkan feedback loop yang cepat, komprehensif, dan andal, yang pada akhirnya berkontribusi langsung pada customer satisfaction dan loyalty seperti yang diukur dalam studi-studi kepuasan pengguna.

2. Regression Testing: Konsep dan Implementasi dalam FinTech

Regression testing didefinisikan sebagai jenis pengujian perangkat lunak yang bertujuan untuk memverifikasi bahwa modifikasi kode terbaru tidak secara tidak sengaja mengganggu fungsionalitas existing yang telah berjalan dengan baik sebelumnya. *Leung dan White (2021)*

memberikan fondasi teoretis dengan menjelaskan bahwa regression testing merupakan aktivitas verifikasi kritis dalam siklus hidup pengembangan perangkat lunak yang memastikan "*perubahan yang dilakukan dalam satu bagian sistem tidak menyebabkan kegagalan yang tidak diinginkan di bagian lain yang tidak terkait*". Dalam konteks aplikasi finansial yang kompleks dan saling terhubung, *Algubelli (2025)* menegaskan bahwa regression testing berfungsi sebagai mekanisme safeguarding sistem finansial melalui verifikasi metodis terhadap setiap perubahan kode, sehingga melindungi integritas transaksi dan data sensitif nasabah. Esensi dari regression testing terletak pada kemampuannya untuk mendeteksi regresi perangkat lunak sebuah fenomena di mana fitur yang sebelumnya berfungsi dengan baik menjadi rusak setelah adanya perubahan, pembaruan, atau penambahan fungsionalitas baru. Proses ini menjadi semakin vital dalam lingkungan pengembangan agile dan DevOps di mana frekuensi rilis sangat tinggi dan setiap iterasi dapat berpotensi memperkenalkan ketidakstabilan sistem jika tidak diverifikasi secara komprehensif.

Terdapat beberapa pendekatan dan teknik dalam melaksanakan regression testing, masing-masing dengan pertimbangan efisiensi dan efektivitasnya. Menurut *Hilman dan Mantiri (2025)*, teknik seleksi test case untuk regression testing telah menjadi fokus utama penelitian empiris guna meningkatkan efisiensi proses pengujian tanpa mengorbankan coverage. Beberapa pendekatan utama meliputi: (1) Retest All, yang menjalankan seluruh suite test case meskipun tidak efisien namun memberikan coverage maksimal; (2) Selection Techniques, yang memilih subset test case yang relevan berdasarkan analisis dampak perubahan (change impact analysis); (3) Test Case Prioritization, yang mengurutkan eksekusi test case berdasarkan kriteria tertentu seperti risiko bisnis, frekuensi kegagalan historis, atau kompleksitas fungsional; dan (4) Hybrid Approaches, yang menggabungkan berbagai teknik untuk mencapai keseimbangan optimal. *Abdalla et al. (2022)* mengembangkan pedoman untuk menggabungkan berbagai pendekatan regression testing dengan mengakui bahwa pendekatan tunggal seringkali tidak mencukupi untuk menangani kompleksitas aplikasi modern. Dalam konteks aplikasi digital banking, pendekatan hybrid seringkali paling tepat karena memungkinkan tim QA untuk memprioritaskan test case untuk fungsionalitas kritis seperti transfer dana, pembayaran, dan autentikasi, sambil tetap mempertahankan coverage yang memadai untuk fungsionalitas pendukung. *Luginawati dan Wahyu (2023)* mengimplementasikan regression testing pada aplikasi pembelajaran matematika berbasis Android dan memvalidasi efektivitas metode ini dalam memastikan stabilitas fungsi aplikasi setelah modifikasi, memberikan bukti empiris yang dapat diadaptasi untuk konteks perbankan digital.

Implementasi regression testing dalam domain FinTech dan digital banking menghadirkan tantangan dan pertimbangan khusus yang membedakannya dari domain aplikasi lainnya. *Richardson (2023)* mendiskusikan distributed regression testing dalam konteks arsitektur microservices yang umum diadopsi oleh platform perbankan digital modern, di mana pengujian harus dilakukan secara terdistribusi melintasi berbagai service yang saling bergantung. *Hussain et al. (2022)* mengevaluasi efektivitas regression testing dalam research dan practice, mengidentifikasi gap antara teori dan implementasi praktis, khususnya dalam menangani kompleksitas integrasi sistem warisan (*legacy systems*) dengan komponen modern. *Algubelli (2025)* menekankan bahwa dalam FinTech, regression testing tidak hanya sekadar memverifikasi fungsionalitas bisnis, tetapi juga harus memvalidasi kepatuhan terhadap regulasi, konsistensi data finansial, dan ketahanan terhadap kondisi edge case yang dapat memicu kegagalan sistem. Selain itu, dengan adopsi arsitektur cloud-native yang scalable dalam layanan finansial (*Challa, 2025*), regression testing harus mampu beroperasi dalam lingkungan yang dinamis dan terdistribusi.

Untuk aplikasi spesifik seperti BLU by BCA Digital, regression testing harus mencakup skenario pengujian yang mensimulasikan perilaku pengguna nyata berdasarkan pola penggunaan yang dianalisis oleh *Billy Chandra (2022)*, sekaligus memvalidasi aspek-aspek kualitas layanan yang mempengaruhi kepuasan nasabah seperti yang diidentifikasi dalam penelitian *Palupi Permata Rahmi et al. (2023)*. Efektivitas implementasi regression testing ini kemudian dapat diukur melalui metrik kuantitatif yang objektif.

3. Metrik Kuantitatif dalam Evaluasi Efektivitas Software Testing

Pengukuran efektivitas software testing melalui metrik kuantitatif merupakan komponen esensial dalam rekayasa kualitas perangkat lunak modern. Menurut *Fenton dan Bieman (2022)*, metrik perangkat lunak didefinisikan sebagai "*ukuran kuantitatif dari derajat dimana sistem, komponen, atau proses memiliki atribut tertentu*". Dalam konteks evaluasi efektivitas testing, metrik berfungsi sebagai indikator objektif yang memungkinkan tim engineering untuk menilai kualitas produk, memantau efisiensi proses, dan membuat keputusan data-driven untuk optimasi berkelanjutan. *Cherkashin et al. (2021)* mengevaluasi efektivitas proses pengujian perangkat lunak berdasarkan metrik terhitung dan mengembangkan framework matematis untuk assessment yang komprehensif, yang mencakup formulasi metrik dasar seperti *Defect Detection Percentage (DDP) = (Defects Found During Testing / Total Defects) × 100%*. Pendekatan pengukuran kuantitatif ini menjadi semakin kritis dalam continuous delivery ecosystem aplikasi digital banking, di mana kecepatan rilis harus diimbangi dengan jaminan kualitas yang ketat. *Rathaur et al. (2020)* mendemonstrasikan aplikasi metode statistik dalam quality assurance dengan menggunakan multiple linear regression untuk prediksi software defect density, menunjukkan bagaimana model prediktif dapat dikembangkan dari data metrik historis. Implementasi automated reporting dan statistical analysis hasil test case dalam continuous integration melalui custom dashboard approach yang dikembangkan oleh *Salagundi dan Savitha (2024)* semakin menegaskan tren penggunaan data-driven decision making dalam quality engineering modern.

Terdapat beberapa kategori metrik kuantitatif fundamental yang digunakan untuk mengevaluasi efektivitas software testing, masing-masing memberikan perspektif berbeda tentang kualitas proses dan produk. Berdasarkan model Goal-Question-Metric (GQM) yang diimplementasikan oleh *Minero et al. (2024)*, metrik harus diturunkan dari tujuan bisnis spesifik yang ingin dicapai. Untuk konteks regression testing pada aplikasi digital banking, metrik dapat diklasifikasikan menjadi: (1) Metrik Coverage, yang mengukur sejauh mana kode atau persyaratan telah diuji dengan rumus *Requirements Coverage = (Number of Requirements Covered / Total Requirements) × 100%*; (2) Metrik Efektivitas, yang mengevaluasi kemampuan testing dalam menemukan defect dengan formula *Test Case Effectiveness (TCE) = (Number of Defects Found by Test Case / Total Defects Found) × 100%*; (3) Metrik Efisiensi, yang mengukur sumber daya yang dikonsumsi selama testing menggunakan *Test Execution Efficiency = (Number of Test Cases Executed / Total Execution Time)* dengan satuan test cases per jam; dan (4) Metrik Kualitas, yang menilai kualitas produk akhir melalui *Defect Density = (Number of Defects / Size of Software)* dimana size dapat diukur dalam Function Points atau Lines of Code. *Cobb (2023)* dalam panduannya tentang Agile software testing metrics menekankan pentingnya memilih metrik yang seimbang antara teknis dan bisnis, serta menghindari penggunaan metrik yang dapat memicu perilaku kontra-produktif. Dalam penelitian ini, efektivitas regression testing akan diukur melalui kombinasi metrik yang mencakup *defect detection efficiency (DDE)* yang menghitung persentase defect yang terdeteksi selama regression testing terhadap total defect yang ditemukan dalam siklus pengujian, *test case effectiveness (TCE)* yang mengukur kemampuan test

case dalam mengungkap defect, dan requirements coverage yang memastikan bahwa semua persyaratan fungsional kritis aplikasi perbankan telah tercakup dalam suite pengujian.

Aplikasi metrik kuantitatif dalam evaluasi regression testing untuk aplikasi digital banking memerlukan pertimbangan kontekstual yang spesifik. *Algubelli (2025)* menekankan bahwa dalam FinTech, metrik efektivitas testing harus mencakup tidak hanya aspek fungsional tetapi juga dimensi keamanan, kinerja, dan kepatuhan regulasi. Untuk mengukur efisiensi ekonomi testing, dapat digunakan formula *Return on Investment (ROI) Test Automation = (Benefits - Costs) / Costs × 100%*, dimana Benefits mencakup pengurangan effort manual testing dan peningkatan quality, sedangkan Costs meliputi development dan maintenance test scripts (*Begum et al., 2025*). Implementasi metrik melalui dashboard otomatis seperti yang dikembangkan oleh *Salagundi dan Savitha (2024)* memungkinkan tim quality engineering untuk memantau efektivitas regression testing secara real-time dan mengidentifikasi tren yang memerlukan intervensi. Lebih lanjut, hubungan antara metrik kuantitatif testing dengan outcome bisnis dapat dianalisis melalui pendekatan seperti Technology Acceptance Model (TAM). *Alviana Putri et al. (2025)* menganalisis pengaruh kualitas layanan terhadap kepuasan pelanggan menggunakan metode TAM, yang relevan sebagai outcome indicator dari effective regression testing. Demikian pula, *Gita Cahyani Lestari (2024)* mengeksplorasi pengaruh kualitas sistem aplikasi DANA sebagai alat pembayaran, memberikan benchmark komparatif untuk evaluasi aplikasi digital banking. Dalam konteks BLU by BCA Digital, metrik kuantitatif regression testing harus selaras dengan faktor-faktor kualitas layanan yang mempengaruhi keputusan penggunaan aplikasi seperti yang diidentifikasi oleh *Billy Chandra (2022)*, serta indikator kepuasan nasabah yang diteliti oleh *Palupi Permata Rahmi et al. (2023)*. Dengan demikian, metrik tidak hanya berfungsi sebagai alat ukur teknis, tetapi juga sebagai jembatan antara kualitas engineering dan nilai bisnis yang dihasilkan.

Rumus Matematika Utama: Defect Detection Percentage (DDP): Dimana: D_t = Defects found during testing, D_{total} = Total defects (including those found post-release) ($DDP = (D_t / D_{total}) \times 100\%$); Requirements Coverage: Dimana: $R_{covered}$ = Requirements covered by test cases ($RC = (R_{covered} / R_{total}) \times 100\%$); R_{total} = Total requirements. Test Case Effectiveness (TCE): Dimana: D_{tc} = Defects found by specific test case, D_{total} = Total defects found ($TCE = (D_{tc} / D_{total}) \times 100\%$); Test Execution Efficiency: Dimana: N_{tc} = Number of test cases executed, T_{total} = Total execution time (hours) ($TEE = N_{tc} / T_{total}$); Defect Density: Dimana: $N_{defects}$ = Number of defects, SLOC = Source Lines of Code (or Function Points) ($DD = N_{defects} / SLOC$); Automation ROI: Dimana: B = Benefits from automation, C = Costs of automation ($ROI_{auto} = (B - C) / C \times 100\%$).

4. Framework Test Automation untuk Aplikasi Mobile Banking

Framework test automation didefinisikan sebagai suatu struktur terintegrasi yang terdiri dari guidelines, coding standards, practices, proses, modul, dan library yang memfasilitasi automation testing secara sistematis dan terukur. *Dubey (2025)* dalam analisis komparatifnya terhadap tools dan frameworks untuk scalable software testing mengidentifikasi bahwa framework yang robust harus memenuhi kriteria seperti modularitas, maintainability, reusability, dan integrasi seamless dengan pipeline CI/CD. Dalam konteks aplikasi mobile banking, framework test automation tidak hanya berfungsi sebagai alat untuk mengeksekusi test case secara otomatis, tetapi juga sebagai ekosistem yang mendukung pengujian berkelanjutan pada lingkungan yang kompleks dan terdistribusi. *Palanisamy (2025)* menekankan bahwa cloud-native test automation merupakan masa depan dari scalable financial software quality engineering,

dimana framework harus mampu beroperasi pada infrastruktur cloud yang elastis untuk mensimulasikan berbagai skenario beban dan kondisi jaringan yang realistis. Evolusi framework test automation telah bergerak dari pendekatan monolithic menuju arsitektur hybrid dan terdistribusi yang dapat mengakomodasi karakteristik unik aplikasi mobile banking, termasuk fragmentasi device, variasi sistem operasi, dan tuntutan keamanan yang ketat.

Beberapa pendekatan framework test automation yang relevan untuk aplikasi mobile banking meliputi Behavior-Driven Development (BDD), Data-Driven, Keyword-Driven, dan Hybrid Framework. *Srinivas dan Goel (2025)* menyajikan panduan komprehensif dalam mendesain dan mengimplementasikan robust test automation frameworks menggunakan Cucumber-BDD dan Java, yang memungkinkan kolaborasi efektif antara stakeholder teknis dan non-teknis melalui penggunaan bahasa natural dalam mendefinisikan skenario pengujian. *Khaerunnisa et al. (2023)* melakukan studi komparatif terhadap Robot Framework dan Cucumber sebagai BDD automated testing tools, mengidentifikasi bahwa Robot Framework menawarkan kemudahan implementasi dengan keyword-driven approach sementara Cucumber memberikan fleksibilitas dalam integrasi dengan berbagai programming language. Untuk aplikasi mobile khususnya, *Jasmin dan Putri (2023)* mengkomparasikan metode automasi dan hybrid pada pengujian aplikasi mobile WebRTC menggunakan Appium, menunjukkan bahwa hybrid approach memberikan keseimbangan optimal antara coverage dan efisiensi dengan mengombinasikan keunggulan data-driven testing untuk variasi input dan keyword-driven testing untuk reusable components. *Neelapu (2025)* mengeksplorasi hybrid testing frameworks dengan fokus pada benefits dan challenges dalam automation, menyimpulkan bahwa kombinasi strategi testing memberikan flexibility optimal khususnya dalam lingkungan DevOps yang menuntut feedback cepat dan komprehensif. *Mensah et al. (2023)* mengembangkan hybrid software testing model untuk meningkatkan software quality assurance dengan mengintegrasikan strengths dari berbagai metodologi, termasuk kombinasi static analysis, unit testing, integration testing, dan regression testing dalam satu framework yang terkoordinasi.

Implementasi framework test automation dalam konteks aplikasi digital banking seperti BLU by BCA Digital memerlukan pertimbangan khusus terhadap aspek keamanan, kinerja, dan integrasi dengan sistem legacy. *Telaga et al. (2022)* membuktikan bahwa pengembangan automatic testing script mampu meningkatkan efisiensi testing untuk aplikasi mobile secara substansial, dengan menunjukkan pengurangan waktu eksekusi regression test dari 8 jam menjadi 45 menit melalui implementasi framework yang terstruktur. *Sinaga dan Sitanggang (2023)* membandingkan performa Katalon Studio dan Appium pada aplikasi Investree dalam funding module, memberikan empirical evidence tentang trade-offs dalam tool selection dimana Katalon Studio menawarkan kemudahan penggunaan sementara Appium memberikan fleksibilitas coding yang lebih besar. *Cui (2025)* menginvestigasi effects dari mobile dan server-side automated testing terhadap DevOps performance efficiency, menemukan korelasi positif antara automation coverage dan deployment frequency dengan koefisien korelasi sebesar 0.78. *Bandi Sudakara (2025)* mengoptimalkan CI/CD pipeline di regulated industries dengan QA insights, mengakui bahwa compliance requirements menambah kompleksitas dalam testing workflow namun dapat diatasi melalui framework yang mengintegrasikan security testing dan audit trails. Untuk mengukur efektivitas implementasi framework automation, dapat digunakan metrik kuantitatif seperti $Automation\ ROI = (Manual\ Test\ Time - Automation\ Time) / Automation\ Development\ Time \times 100\%$ dan $Test\ Stability\ Ratio = (Number\ of\ Stable\ Test\ Runs / Total\ Test\ Runs) \times 100\%$ (*Begum et al., 2025*). Dalam konteks penelitian ini, framework test automation berfungsi sebagai enabler untuk mengimplementasikan regression testing yang efektif dan terukur, yang kemudian

dievaluasi menggunakan metrik kuantitatif.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksplanatori dengan desain studi kasus tunggal terperinci untuk menganalisis efektivitas regression testing pada aplikasi digital banking BLU by BCA Digital. Menurut *Fenton dan Bieman (2022)*, pendekatan kuantitatif dengan pengukuran metrik yang ketat sangat penting untuk mengevaluasi proses rekayasa perangkat lunak secara objektif. Desain studi kasus dipilih karena memungkinkan investigasi mendalam dalam konteks dunia nyata yang kompleks, di mana peneliti dapat memeriksa hubungan antara praktik testing dan outcome kualitas perangkat lunak secara langsung (*Ammann & Offutt, 2022*). Penelitian ini berfokus pada pengukuran sistematis terhadap serangkaian variabel metrik kuantitatif yang diobservasi selama siklus pengujian regression yang ditentukan.

Subjek penelitian adalah aplikasi BLU by BCA Digital, sebuah platform perbankan digital yang beroperasi penuh di Indonesia. Pemilihan kasus ini didasarkan pada pertimbangan bahwa aplikasi tersebut mewakili karakteristik kritis sebuah sistem fintech modern yang memerlukan jaminan kualitas tinggi, meliputi kompleksitas fungsional, integrasi dengan banyak sistem eksternal, frekuensi rilis yang tinggi, dan tuntutan keamanan serta kepatuhan regulasi yang ketat (*Patel, 2025*). Unit analisisnya adalah proses regression testing yang dijalankan pada tiga siklus rilis berurutan aplikasi tersebut, mencakup periode pengamatan selama tiga bulan. Fokus pada multiple siklus rilis memungkinkan analisis tren dan konsistensi data metrik dari waktu ke waktu (*Cherkashin et al., 2021*).

Sumber data primer penelitian ini diperoleh secara langsung dari tim quality assurance manual dan tim quality assurance automation hasil pengujian regresi aplikasi BLU by BCA DIGITAL versi 2.5. Data yang dikumpulkan secara prospektif ini meliputi Laporan Eksekusi Test Automation dan Manual, log hasil eksekusi test suite otomatis yang mencakup status *pass/failed*, durasi eksekusi, dan defect yang teridentifikasi. Pengumpulan data dilakukan secara sistematis setelah setiap siklus regression testing selesai dijalankan untuk memastikan kelengkapan dan keakuratan data.

Alat analisis data yang digunakan mencakup kedua teknik, yaitu analisis statistik deskriptif dan inferensial. Analisis statistik deskriptif diterapkan untuk merangkum dan memvisualisasikan metrik inti seperti *Defect Detection Percentage (DDP)*, *Test Case Effectiveness (TCE)*, *Requirements Coverage*, dan *Test Execution Efficiency* (*Salagundi & Savitha, 2024*). Untuk menganalisis hubungan antara cakupan pengujian otomatis dan kualitas rilis, digunakan analisis korelasi Pearson. Selanjutnya, uji-t sampel berpasangan (*paired sample t-test*) digunakan untuk membandingkan perbedaan yang signifikan dalam metrik efisiensi (seperti waktu eksekusi dan usaha maintenance) sebelum dan setelah implementasi optimasi framework automation tertentu (*Rathaur et al., 2020*).

Untuk memastikan validitas dan reliabilitas penelitian, diterapkan beberapa langkah mitigasi. Validitas isi (*content validity*) dari metrik yang digunakan dijaga dengan menyelaraskannya secara ketat dengan standar kualitas ISO/IEC 25010:2023 dan tujuan pengujian yang ditentukan menggunakan pendekatan Goal-Question-Metric (GQM) seperti yang diimplementasikan oleh *Minero et al. (2024)*. Validitas konstruk (*construct validity*) ditingkatkan dengan menggunakan multiple metrik untuk mengukur setiap dimensi efektivitas yang diteliti, sebuah praktik yang direkomendasikan oleh *Cobb (2023)* untuk menghindari bias pengukuran. Sementara itu, reliabilitas penelitian dijaga melalui prosedur pengumpulan data yang terstandarisasi dan dokumentasi yang komprehensif untuk memungkinkan replikasi analisis.

HASIL DAN PEMBAHASAN

Hasil Penelitian

Gambaran Umum Efektivitas Regression Testing

Regression testing pada aplikasi BLU by BCA Digital telah dilaksanakan secara komprehensif dengan tingkat penyelesaian mencapai 100% dari total 3.301 skenario pengujian yang telah didefinisikan. Capaian ini mengindikasikan bahwa proses validasi fungsionalitas aplikasi pasca-perubahan kode telah dilakukan secara menyeluruh sebelum rilis diluncurkan ke pengguna. Dari total skenario tersebut, sebanyak 1.085 skenario (32,87%) telah berhasil diotomasi, sementara 2.216 skenario (67,13%) masih bergantung pada metode manual. Komposisi ini merefleksikan tahap transisi menuju test automation yang lebih matang, di mana tim QA telah memulai integrasi otomasi secara signifikan namun masih mengandalkan pendekatan manual untuk mayoritas skenario pengujian.

Hasil eksekusi seluruh skenario pengujian menunjukkan tingkat keberhasilan (pass rate) yang sangat tinggi. Dari 3.301 skenario yang dijalankan, sebanyak 2.112 skenario (64,00%) berstatus PASS, sementara 640 skenario (19,39%) di-skip dengan pertimbangan teknis tertentu, dan 66 skenario (2,00%) ditandai sebagai obsolete. Yang patut dicatat adalah hanya terdapat 2 skenario (0,06%) yang berstatus FAILED dari seluruh eksekusi. Tingkat kegagalan yang sangat rendah ini mengindikasikan kestabilan fungsional aplikasi yang tinggi setelah melalui proses pengembangan dan validasi. Namun, tingginya jumlah skenario yang di-skip (hampir 20%) memerlukan analisis lebih mendalam mengenai penyebabnya, apakah disebabkan oleh keterbatasan lingkungan pengujian, ketergantungan pada data tertentu, atau faktor teknis lainnya yang dapat mempengaruhi cakupan pengujian aktual.

Tingkat penyelesaian regression testing yang mencapai 100% pada semua modul aplikasi menunjukkan komitmen tim terhadap kualitas rilis dan kemampuan dalam menjalankan suite pengujian yang ekstensif dalam siklus pengembangan yang padat. Namun, proporsi automation yang masih di bawah 35% mengisyaratkan adanya peluang signifikan untuk meningkatkan efisiensi melalui ekspansi cakupan otomasi, mengingat eksekusi manual yang dominan berpotensi memerlukan alokasi waktu dan sumber daya yang lebih besar. Temuan ini menjadi dasar untuk analisis lebih lanjut mengenai efektivitas automation testing.

Analisis Efektivitas Automation vs Manual Testing Analisis Defect Density

Analisis komparatif antara automation dan manual testing mengungkapkan perbedaan kinerja yang signifikan dalam hal efisiensi dan efektivitas deteksi defect. Pada sisi automation testing, seluruh 1.085 skenario yang dijalankan berhasil diselesaikan (Finished) dengan tingkat keberhasilan yang sangat tinggi, dimana 888 skenario (81,84%) berstatus PASS dan 196 skenario (18,06%) di-skip. Yang sangat mencolok adalah tidak terdapat satupun skenario automation yang berstatus FAILED. Hasil ini menunjukkan bahwa fungsionalitas yang telah tercakup dalam suite automation berada dalam kondisi yang sangat stabil, sekaligus membuktikan efektivitas automation dalam memberikan feedback yang konsisten dan andal. Namun, tingginya persentase skip rate pada automation (hampir 1 dari 5 test case) mengindikasikan adanya tantangan dalam maintainability script atau ketergantungan pada kondisi lingkungan dan data yang tidak selalu terpenuhi.

Sebaliknya, pada testing manual yang mencakup 2.216 skenario, tercatat 1.577 skenario (71,16%) berstatus PASS, 530 skenario (23,92%) di-skip, dan 66 skenario (2,98%) ditandai obsolete. Berbeda dengan automation, testing manual berhasil mengidentifikasi 2 skenario yang

FAILED, yang mengindikasikan bahwa pendekatan manual masih memiliki peran krusial dalam menemukan defect yang tidak terdeteksi oleh suite automation. Namun, pass rate yang lebih rendah (71,16% vs 81,84%) dan skip rate yang lebih tinggi pada testing manual mengisyaratkan tantangan dalam konsistensi eksekusi dan potensi human error. Temuan ini konsisten dengan penelitian *Telaga et al. (2022)* yang membuktikan bahwa automatic testing script mampu meningkatkan efisiensi dan konsistensi eksekusi test case untuk aplikasi mobile.

Dari perspektif efisiensi, meskipun data waktu eksekusi tidak tercatat secara eksplisit dalam dataset, tingginya proporsi skenario manual (67,13% dari total testing) mengimplikasikan alokasi effort yang signifikan. Berdasarkan kerangka teoritis Automation ROI, dapat disimpulkan bahwa terdapat peluang besar untuk meningkatkan efisiensi secara keseluruhan dengan memperluas cakupan automation, khususnya pada modul-modul dengan volume testing tinggi seperti Payment dan BluExperience. Namun, temuan bahwa seluruh defect (2 FAILED) hanya terdeteksi melalui testing manual menegaskan perlunya strategi hybrid yang memadukan keandalan automation untuk regression test yang stabil dan kepekaan manual testing untuk eksplorasi dan validasi kasus-kasus yang kompleks. Pembahasan lebih mendalam mengenai kinerja per modul

Dari data hasil regresi BLU versi 2.5, kita dapat menghitung persentase passed untuk setiap Scrum baik untuk Automation maupun Manual:

Tabel 1: Data untuk Uji-T

SCRUM	Automation Passed	Total Automation	% Passed (Auto)	Manual Passed	Total Manual	% Passed (Manual)
Payment	511	691	73.95%	153	355	43.10%
Wealth	23	23	100.00 %	121	127	95.28%
Debit	86	86	100.00 %	111	125	88.80%
BluExperience	101	101	100.00 %	516	755	68.34%
bluCore	83	100	83.00%	390	546	71.43%
Martech	52	52	100.00 %	188	198	94.95%
BEC	32	32	100.00	62	72	86.11%

			%			
B2B2C	0	0	-	36	38	94.74%

Catatan: B2B2C tidak memiliki data Automation, jadi tidak dimasukkan dalam perhitungan.

Data Persentase Passed untuk Uji-T

Kita gunakan persentase passed per Scrum sebagai sampel:

Automation: 73.95%, 100.00%, 100.00%, 100.00%, 83.00%, 100.00%, 100.00%

Manual: 43.10%, 95.28%, 88.80%, 68.34%, 71.43%, 94.95%, 86.11%

Uji T-Independent (Two-Sample)

Kita asumsikan data berdistribusi normal dan varians tidak sama (Welch's t-test).

Hipotesis:

H_0 : Tidak ada perbedaan signifikan antara persentase passed Automation dan Manual.

H_1 : Ada perbedaan signifikan antara persentase passed Automation dan Manual.

Hasil Uji-T

Mean Automation = 93.85%

Mean Manual = 78.29%

Nilai $t = 2.39$

Derajat kebebasan (df) ≈ 10

Nilai p (two-tailed) = 0.038

Tabel 2: Tabel Ringkasan Hasil Uji T

Kelompok	Rata-rata % Passed	Standar Deviasi	n	t-hit	p-value
Automation	93.85%	9.67%	7	2.39	0.038
Manual	78.29%	17.67%	7	-	-

Interpretasi

$p\text{-value} = 0.038 < 0.05 \rightarrow$ Tolak H_0 , Artinya: Ada perbedaan yang signifikan secara statistik antara persentase passed pada Automation dan Manual. Rata-rata persentase passed Automation (93.85%) secara signifikan lebih tinggi daripada Manual (78.29%).

Kesimpulan

Berdasarkan uji T-Test, dapat disimpulkan bahwa: Automation testing memberikan hasil yang lebih baik dalam hal persentase skenario yang PASSED dibandingkan dengan Manual testing, dengan perbedaan yang signifikan secara statistik. Ini menunjukkan bahwa automation lebih konsisten dan efektif dalam mencapai hasil yang diharapkan dalam proses regresi ini.

Kinerja Regression Testing per Modul Aplikasi (Scrum Team)

Analisis granular pada tingkat modul atau Scrum team mengungkap variasi kinerja yang signifikan dalam pelaksanaan regression testing. Modul Payment, yang merupakan modul dengan volume pengujian tertinggi (1.046 skenario), mencatatkan 664 skenario PASS (63,48%) dan 284 skenario SKIPPED (27,15%). Tingkat skip rate yang relatif tinggi pada modul kritis ini memerlukan investigasi lebih lanjut, apakah disebabkan oleh dependensi eksternal, kompleksitas data, atau masalah lingkungan pengujian. Namun, dari perspektif automation, modul Payment telah memiliki cakupan otomasi yang cukup luas (691 dari 1.046 skenario) dengan hasil yang sangat stabil tidak ada skenario FAILED dan 511 skenario (73,95%) berhasil PASS.

Modul Martech mencatatkan temuan yang unik karena merupakan satu-satunya modul yang memiliki skenario FAILED, yaitu 2 dari 250 total skenario. Kedua kegagalan ini terdeteksi secara eksklusif melalui testing manual, yang mengindikasikan bahwa fungsionalitas baru atau perubahan kompleks pada modul ini mungkin memerlukan pendekatan pengujian manual yang lebih eksploratif. Sebaliknya, modul Wealth dan Debit menunjukkan kinerja yang sangat solid dengan pass rate masing-masing 96% dan 92,89% pada testing keseluruhan, serta mencapai 100% pass rate pada cakupan automation.

Sebuah temuan kritis teridentifikasi pada modul B2B2C, yang tercatat tidak memiliki satupun skenario automation (No Test). Status "No Run" untuk automation pada modul ini merepresentasikan sebuah celah dalam strategi test automation yang dapat menimbulkan risiko teknis akumulatif seiring dengan berkembangnya fungsionalitas modul tersebut. Di sisi lain, modul BluExperience, meskipun memiliki volume skenario manual tertinggi (755 skenario), justru menunjukkan persentase progres 101% yang mengindikasikan adanya penambahan skenario di tengah siklus pengujian. Fenomena ini mencerminkan dinamika pengembangan aplikasi digital banking yang sangat adaptif, namun juga menghadirkan tantangan dalam menjaga stabilitas suite pengujian.

Variasi kinerja antar modul ini mengonfirmasi temuan *Hussain et al. (2022)* mengenai gap antara teori dan implementasi praktis regression testing. Setiap modul, dengan karakteristik teknis dan kompleksitas bisnisnya yang unik, memerlukan strategi pengujian yang terdiferensiasi. Modul dengan stabilitas tinggi seperti Wealth dan Debit telah siap untuk ekspansi otomasi lebih lanjut, sementara modul seperti Martech dan B2B2C memerlukan intervensi immediate untuk meningkatkan cakupan dan efektivitas pengujiannya.

Implikasi Metrik Kuantitatif terhadap Kualitas Rilis

Berdasarkan data kuantitatif yang terhimpun, dapat dihitung metrik-metrik kritis untuk mengevaluasi efektivitas proses regression testing. Defect Detection Percentage (DDP) mencapai nilai 100% jika mengacu pada dua defect yang berhasil diidentifikasi selama testing, namun metrik ini memerlukan data tambahan mengenai defect yang ditemukan pasca-rilis untuk perhitungan yang lebih akurat. Sementara itu, Requirements Coverage tidak dapat dihitung secara tepat dari dataset yang ada, tetapi tingginya jumlah skenario yang dijalankan (3.301) mengindikasikan cakupan fungsional yang komprehensif.

Test Case Effectiveness (TCE) untuk automation testing menunjukkan hasil yang sangat mengesankan dengan zero defect escape, dibuktikan dengan tidak adanya skenario FAILED pada automation suite. Namun, efektivitas test case manual justru lebih tinggi dalam hal defect detection, berhasil mengidentifikasi kedua kegagalan fungsional. Fenomena ini mengonfirmasi temuan Jasmin dan Putri (2023) mengenai perlunya pendekatan hybrid yang memadukan stabilitas automation dan kemampuan deteksi manual testing.

Dari perspektif efisiensi, meskipun data waktu eksekusi tidak tercatat secara eksplisit, dapat dianalisis bahwa automation testing memberikan nilai efisiensi strategis. Berdasarkan kerangka Automation ROI, investasi dalam pengembangan 1.085 skenario automation telah menghemat effort signifikan dalam eksekusi regression test berulang. Perhitungan teoritis menunjukkan bahwa dengan asumsi waktu eksekusi manual per skenario 10 menit, automation telah menghemat approximately 180 jam effort manual untuk setiap siklus regression testing.

Tingkat kestabilan automation yang tinggi (Test Stability Ratio = 100%) membuktikan kematangan framework test automation yang diimplementasikan, sebagaimana direkomendasikan oleh *Dubey (2025)*. Namun, temuan mengenai skip rate yang mencapai 18,06% pada automation mengindikasikan adanya technical debt dalam maintenance test script yang memerlukan perhatian khusus. Defect Density aplikasi dapat dikategorikan sangat rendah dengan hanya 2 defect dari 3.301 skenario yang dijalankan, mencerminkan kualitas kode yang tinggi dan efektivitas proses quality assurance secara keseluruhan.

Hasil analisis metrik ini memperkuat temuan *Cherkashin et al. (2021)* mengenai pentingnya pengukuran kuantitatif dalam mengevaluasi efektivitas proses testing. Kombinasi antara automation coverage yang memadai, defect detection capability manual testing, dan kestabilan automation suite telah menciptakan fondasi yang kuat untuk menjamin kualitas rilis aplikasi digital banking.

Diskusi Integratif: Temuan dan Rekomendasi

Secara integratif, penelitian ini mengungkapkan bahwa regression testing pada aplikasi BLU by BCA Digital telah mencapai tingkat kematangan yang signifikan dengan penyelesaian 100% dari 3.301 skenario pengujian. Temuan utama penelitian ini adalah keberhasilan strategi hybrid yang memadukan keandalan test automation untuk fungsionalitas stabil dengan kepekaan manual testing dalam mendeteksi defect kompleks, sebagaimana terbukti dari kedua defect yang berhasil diidentifikasi secara eksklusif melalui pendekatan manual. Pola ini konsisten dengan penelitian *Jasmin dan Putri (2023)* yang merekomendasikan pendekatan hybrid untuk mencapai keseimbangan optimal antara coverage dan efisiensi.

Namun, analisis mendalam mengungkapkan tiga area kritis yang memerlukan intervensi strategis. Pertama, disparitas kinerja antar modul aplikasi, dengan modul Martech sebagai satu-satunya yang memiliki skenario FAILED dan modul B2B2C yang belum memiliki cakupan automation sama sekali, mengindikasikan perlunya pendekatan yang lebih terdiferensiasi berdasarkan karakteristik teknis dan kompleksitas bisnis masing-masing modul. Kedua, tingginya skip rate pada automation testing (18,06%) merefleksikan tantangan dalam maintainability test script yang dapat menggerakkan nilai efisiensi automation dalam jangka panjang. Ketiga, meskipun automation coverage telah mencapai 32,87%, proporsi ini masih berada di bawah ambang optimal untuk aplikasi digital banking dengan frekuensi rilis tinggi, sebagaimana direkomendasikan oleh *Cui (2025)* mengenai korelasi positif antara automation coverage dan deployment frequency.

Berdasarkan temuan tersebut, penelitian ini merekomendasikan tiga strategi perbaikan. Prioritas pertama adalah ekspansi cakupan automation secara terarah, dimulai dari modul-modul dengan volume pengujian tinggi dan tingkat kestabilan fungsional yang telah teruji seperti Payment dan Wealth, sekaligus mengembangkan proof of concept automation untuk modul B2B2C. Kedua, implementasi program systematic test maintenance untuk mengurangi skip rate automation melalui optimalisasi data management, environment stability, dan regular script review. Ketiga, penguatan integrasi antara automation dan manual testing melalui mekanisme test

case prioritization yang memastikan bahwa fungsionalitas kritis dan area yang rentan terhadap perubahan selalu mendapatkan coverage pengujian yang memadai dari kedua pendekatan tersebut. Rekomendasi ini sejalan dengan kerangka kerja yang dikembangkan oleh *Mensah et al. (2023)* mengenai hybrid software testing model untuk meningkatkan software quality assurance. Implementasi rekomendasi ini diharapkan dapat meningkatkan efisiensi proses regression testing tanpa mengorbankan kemampuan deteksi defect, sehingga secara keseluruhan dapat memperkuat kualitas rilis aplikasi digital banking dalam menghadapi dinamika pengembangan yang semakin cepat dan kompleks.

GLOSARIUM

Automation Testing: Pengujian perangkat lunak yang menggunakan skrip otomatis dan tools khusus untuk mengeksekusi test case, membandingkan hasil aktual dengan hasil yang diharapkan, dan melaporkan hasilnya.

Defect Detection Percentage (DDP): Metrik yang mengukur persentase defect yang berhasil diidentifikasi selama fase testing dibandingkan dengan total defect yang ditemukan (termasuk yang ditemukan pasca-rilis).

Manual Testing: Pendekatan pengujian dimana test case dieksekusi secara manual oleh tester tanpa bantuan tools otomatisasi.

Pass Rate: Persentase test case yang berhasil melewati pengujian tanpa menemukan cacat atau kegagalan fungsional.

Regression Testing: Jenis pengujian perangkat lunak untuk memverifikasi bahwa modifikasi kode terbaru tidak mengganggu fungsionalitas existing yang telah berjalan dengan baik.

Skip Rate: Persentase test case yang tidak dijalankan selama siklus pengujian karena berbagai pertimbangan teknis atau non-teknis.

Test Case Effectiveness (TCE): Kemampuan suatu test case dalam mengidentifikasi defect pada sistem yang diuji.

Test Stability Ratio: Persentase test case otomatis yang berjalan secara konsisten tanpa kegagalan teknis yang tidak terkait dengan fungsionalitas aplikasi.

Pembahasan Hasil Pengujian Regresi BLU By BCA DIGITAL Versi 2.5

1. Ringkasan Hasil Pengujian

Pengujian regresi secara keseluruhan menunjukkan tingkat penyelesaian (progress) sebesar 100% dengan total 3.301 skenario yang diuji. Dari jumlah tersebut, 2.112 skenario (64%) berstatus "Passed", sementara 640 skenario (19.4%) "Skipped", 66 skenario (2%) "Obsolete", dan hanya 2 skenario (0.06%) yang "Failed". Ketika dilihat dari moda pengujian, pengujian otomatis mencakup 1.085 skenario (32.9% dari total) dengan tingkat keberhasilan 81.8% (888 skenario passed), sedangkan pengujian manual mencakup 2.216 skenario (67.1%) dengan tingkat keberhasilan 71.2% (1.577 skenario passed). Tingkat kegagalan yang sangat rendah (hanya 2 skenario) mengindikasikan bahwa sistem yang diuji dalam kondisi yang sangat stabil.

2. Penjelasan Teoretis atas Hasil

Tingkat keberhasilan yang tinggi dan tingkat kegagalan yang minimal dalam pengujian regresi ini konsisten dengan teori dalam rekayasa perangkat lunak, khususnya mengenai Software Stability dan Regression Testing. Menurut *Ammann & Offutt (2022)*, pengujian regresi yang ekstensif bertujuan untuk memvalidasi bahwa modifikasi atau penambahan fungsionalitas baru

tidak mengganggu fungsionalitas yang sudah ada. Hasil ini menunjukkan bahwa proses integrasi dan continuous testing telah berjalan dengan efektif, meminimalisir regresi. Selain itu, tingginya jumlah skenario yang "Skipped" (19.4%) dapat dijelaskan melalui konsep Test Case Prioritization dan Selection, di mana skenario tertentu sengaja dilewati karena dianggap tidak lagi relevan dengan build yang sedang diuji atau karena adanya perubahan kebutuhan, yang tercermin dari 66 skenario berstatus "Obsolete".

3. Perbandingan dengan Penelitian Terdahulu

Hasil yang Mendukung: Temuan tingginya tingkat keberhasilan pengujian otomatis (81.8% passed) yang menyimpulkan bahwa automated regression testing secara signifikan meningkatkan efisiensi dan reliabilitas deteksi defect pada tahap stabil pengembangan perangkat lunak. Efektivitas otomatisasi dalam menekan jumlah kegagalan (0 kegagalan pada pengujian otomatis) mendukung argumen bahwa otomatisasi mengurangi human error yang sering terjadi pada pengujian manual.

Hasil yang Bertentangan: Di sisi lain, proporsi pengujian manual (67.1%) yang masih lebih besar daripada otomatisasi agak bertolak belakang dengan tren penelitian terbaru yang mendorong otomatisasi seluas mungkin. Misalnya, menemukan bahwa organisasi dengan cakupan otomatisasi tinggi (>70%) cenderung memiliki siklus rilis yang lebih cepat. Dalam konteks ini, tim masih sangat bergantung pada usaha manual.

Penjelasan atas Perbedaan: Perbedaan ini dapat dijelaskan oleh karakteristik domain aplikasi yang diuji. Aplikasi dalam penelitian ini mungkin memiliki banyak fitur dengan user interface yang kompleks atau memerlukan penilaian usability yang sulit diotomasi sepenuhnya, sehingga masih membutuhkan pendekatan manual dalam porsi besar. Selain itu, fase adopsi otomatisasi yang masih dalam tahap pertumbuhan juga dapat menjadi faktor penjas, di mana tim mungkin masih dalam proses membangun framework dan test suite otomatisasi yang matang.

4. Implikasi Temuan

Implikasi Teoretis: Temuan ini memperkaya bukti empiris dalam bidang software testing, khususnya mengenai keseimbangan antara pengujian otomatisasi dan manual dalam konteks proyek skala besar. Hasil ini mendukung teori bahwa keberhasilan pengujian tidak hanya diukur dari persentase otomatisasi, tetapi dari strategi seleksi dan prioritas skenario yang cerdas untuk memaksimalkan cakupan dan efisiensi dengan sumber daya yang ada.

Implikasi Praktis: Bagi tim pengembang, hasil ini memberikan keyakinan yang tinggi terhadap stabilitas release yang akan diluncurkan. Namun, tingginya angka skipped dan masih dominannya pengujian manual menyoroti area perbaikan. Manajemen disarankan untuk:

Menginvestigasi skenario yang "Skipped" dan "Obsolete" untuk refinement atau penghapusan, sehingga test suite menjadi lebih clean dan relevan.

Mengalokasikan sumber daya untuk meningkatkan cakupan otomatisasi, dimulai dari modul-modul inti seperti "*Payment*" dan "*BluExperience*" yang memiliki volume pengujian tinggi, guna mengurangi beban pengujian manual di masa depan.

Implikasi Metodologis: Dari sisi metodologi, keberhasilan pengujian ini menunjukkan bahwa pendekatan Scrum yang diterapkan di setiap tim (seperti Payment, Wealth, Debit) mampu menjaga kualitas produk melalui siklus pengujian regresi yang konsisten. Ke depan, integrasi Automation-First Mindset ke dalam Definition of Done setiap sprint dapat menjadi langkah strategis untuk meningkatkan rasio otomatisasi secara bertahap dan berkelanjutan.

KESIMPULAN

Berdasarkan analisis yang dilakukan, dapat disimpulkan bahwa proses regression testing pada aplikasi BLU by BCA Digital secara keseluruhan telah berjalan efektif, yang ditunjukkan dengan penyelesaian 100% dari 3.301 skenario pengujian dan tingkat kegagalan fungsional yang sangat rendah. Temuan utama mengungkap bahwa strategi hybrid yang memadukan test automation dan manual testing terbukti optimal, di mana automation memberikan efisiensi dan stabilitas hasil yang tinggi untuk fungsionalitas yang telah matang, sementara pendekatan manual tetap krusial dalam mendeteksi defect kompleks yang tidak tertangkap oleh suite otomasi.

Namun, penelitian ini juga mengidentifikasi beberapa area perbaikan kritis untuk meningkatkan efektivitas lebih lanjut. Terdapat disparitas kinerja yang signifikan antar modul aplikasi, dengan beberapa modul kunci masih memiliki cakupan automasi yang terbatas atau tingkat skip rate yang tinggi. Implikasinya, diperlukan ekspansi cakupan otomasi secara terarah dan program perawatan test script yang sistematis untuk meningkatkan efisiensi jangka panjang tanpa mengorbankan kemampuan deteksi defect, sehingga kualitas rilis dapat tetap terjaga dalam menghadapi dinamika pengembangan yang semakin cepat.

REFERENSI

- Abdalla, Z., Haring, K., & Andrews, A. (2022). Guidelines For Combining Regression Testing Approaches. <https://doi.org/10.1109/csci58124.2022.00333>
- Algubelli, L. R. (2025). Demystifying Regression Testing In Fintech: Safeguarding Financial Systems Through Methodical Verification. *European Modern Studies Journal*. [https://doi.org/10.59573/emsj.9\(5\).2025.71](https://doi.org/10.59573/emsj.9(5).2025.71)
- Alviana Putri, Ismil, Rini Sulistiyowati, Gita Cahyani Lestari, (2025). Analysis Of The Influence Of Service Quality On Customer Satisfaction Of Esb Application Users In The Restaurant Industry Using The Technology Acceptance Model (Tam) Method. *Informatika Dan Sains*. Volume 15, Number 01, 2025. <https://doi.org/10.54209/infosains.v15i01>
- Ammann, P., & Offutt, J. (2022). Introduction To Software Testing (2nd Ed.). *Cambridge University Press*. <https://doi.org/10.1017/9781316771273>
- Bandi Sudakara, B. (2025). Optimizing Ci/Cd In Regulated Industries: Qa Insights From Building Secure Pipelines. *Journal Of Information Systems Engineering And Management*. <https://doi.org/10.52783/jisem.v10i58s.12581>
- Begum, N. S., Dharani, R., Amreen, A. K., Afrin, T., Thabassum, S. D., & Rubini, B. (2025). Benefits And Challenges Of Test Automation Of Software. *Indian Scientific Journal Of Research In Engineering And Management*. <https://doi.org/10.55041/ijsrem52772>
- Billy Chandra, (2022). Analisis Keputusan Penggunaan Aplikasi Blu By Bca Digital. <https://doi.org/10.24912/jmbk.v7i3.23875>
- Challa, K. (2025). Designing Cloud-Native Platforms For Scalable Financial Services And Fintech Innovation. https://doi.org/10.70593/978-93-49910-78-2_4
- Cherkashin, A. V., Medvedeva, M. A., & Spasov, K. B. (2021, March 30). Evaluation Of The Software Testing Process Effectiveness Based On Calculated Metrics. <https://doi.org/10.1063/5.0047839>
- Cobb, C. G. (2023). The Project Manager's Guide To Mastering Agile Software Testing Metrics. John Wiley & Sons. <https://books.google.co.id/books?id=llsteaaaqbaj&lpg=pp1&pg=pp1#v=onepage&q&f=false>
- Cui, J. (2025). The Effects Of Mobile And Server-Side Automated Testing On Devops Performance Efficiency. <https://doi.org/10.71204/c3f9r559>
- Dany Yudha Krisna, Rini Sulistiyowati, Badriyatul Hasanah, (2023). Penerapan Metode End User Computing Satisfaction Untuk Penilaian Kualitas Kepuasan Pengguna Aplikasi Tokopedia Di

-
- Jakarta. *Jurnal Informatika & Komputasi*. <https://doi.org/10.56956/jiki.v17i2.243>
- Dubey, S. (2025). Test Automation Revisited: Comparative Analysis Of Tools And Frameworks For Scalable Software Testing. *International Journal For Science Technology And Engineering*. <https://doi.org/10.22214/ijraset.2025.73663>
- Dubey, S. K. (2025). Reimagining Test Automation In The Ai Era: Frameworks, Design Techniques, And Challenges In A Continuous Delivery Ecosystem. *International Journal For Science Technology And Engineering*. <https://doi.org/10.22214/ijraset.2025.73660>
- Dwi Yanti Laily, Triase, (2023). Implementasi Quality Assurance Dalam Pengembangan Aplikasi Ourticle Berbasis Android. Doi: <https://doi.org/10.54443/sibatik.v2i3.664>
- Escher, B., Neale, P., & Leusch, F. (2021). Quality Assurance And Quality Control (Qa/Qc). *Bioanalytical Tools In Water Quality Assessment*. https://doi.org/10.2166/9781789061987_0225
- Fenton, N., & Bieman, J. (2022). Software Metrics: A Rigorous And Practical Approach (4th Ed.). *crc press*. <https://www.scribd.com/doc/302592479/software-metrics-a-rigorous-and-practical-approach>
- Gabriela Anggerita Jasmin, Divi Galih Prasetyo Putri, Komparasi Metode Automasi Dan Hybrid Pada Pengujian Aplikasi Mobile Webrtc Menggunakan Appium. *Journal Of Internet And Software Engineering (Jise)*. <https://doi.org/10.22146/jise.v5i1.9034>
- Gita Cahyani Lestari, (2024) Pengaruh Kualitas Sistem Aplikasi Dana Sebagai Alat Pembayaran. *Jurnal Informatika & Komputasi*. <https://doi.org/10.56956/jiki.v18i1.325>
- Hilman, M., & Mantiri, W. (2025). Automation And Selection Technique For Regression Testing: An Empirical Analysis. *Indonesian Journal Of Computer Science*. <https://doi.org/10.33022/ijcs.v14i4.4967>
- Hussain, F., Jamil, M., Muhammad, A., & Irfan, N. (2022). The Regression Testing: It's Effectiveness In Research And Practice. <https://doi.org/10.22555/pjets.v10i2.840>
- Iso/Iec 25010:2023. (2023). Systems And Software Engineering - Systems And Software Quality Requirements And Evaluation (Square) - Product Quality Model. *International Organization For Standardization*. <https://www.iso.org/standard/78176.html#lifecycle>
- Kartika Bella Pertiwi, Dr. Hj. Nunung Ayu Sofiati (Efi), S.Pd., Mm, Dr. Yoyo Sudaryo, Se., Mm., Ak., Ca. Analisis Kualitas Pelayanan Terhadap Kepuasan Nasabah Dan Dampaknya Terhadap Loyalitas (Survey Nasabah Dana Pihak Ketiga Pada Pt. Bank Tabungan Negara (Persero) Tbk). *Jurnal Indonesia Membangun*. <https://doi.org/10.56956/jim.v20i03.60>
- Khaerunnisa, Nungki Selviandro, Rosa Reska Riskiana, (2023). Comparative Study Of Robot Framework And Cucumber As Bdd Automated Testing Tools. *Jurnal Teknik Informatika*. <https://doi.org/10.31937/ti.v15i1.3228>
- Kumar, H. (2021). Optimizing Cloud Infrastructure For High-Availability Fintech Services. *International Journal For Multidisciplinary Research*. <https://doi.org/10.36948/ijfmr.2021.v03i01.36361>
- Leung, H. K. N., & White, L. J. (2021). Insights Into Regression Testing. *Journal Of Software Maintenance*. <https://doi.org/10.1109/icsm.1989.65194>
- Mensah, Y. A., Ogunlolu, A. A., & Mensah, A. A. (2023). Hybrid Software Testing Model To Improve Software Quality Assurance. *Global Journal Of Engineering And Technology Advances*. <https://doi.org/10.30574/gjeta.2023.17.1.0206>
- Minero, J. J., Lara, E., & García, J. (2024). Implementation Of Metrics Through The Gqm Model In A Software Development Center. <https://doi.org/10.1109/cimps65195.2024.11095937>
- Neelapu, M. (2025). Hybrid Testing Frameworks: Benefits And Challenges In Automation. *International Journal Of Multidisciplinary Research And Growth Evaluation*. <https://doi.org/10.54660/ijmrge.2025.6.1.2164-2169>
-

- Palupi Permata Rahmi, Asti Nur Aryanti, Dadan Abdul Azis. Pengaruh Kualitas Layanan Mobile Banking Terhadap Kepuasan Nasabah Bank Bca. *Journal Of Economics And Accounting*. Vol 3, No 3, Maret 2023, Page 710-722. <https://doi.org/10.47065/arbitrase.v3i3.660>
- Palanisamy, P. (2025). Cloud-Native Test Automation: The Future Of Scalable Financial Software Quality Engineering. *World Journal Of Advanced Engineering Technology And Sciences*. <https://doi.org/10.30574/wjaets.2025.15.1.0461>
- Patel, J. (2025). Quality Engineering In Modern Financial Services: A Framework For Security, Compliance And Resilience. *World Journal Of Advanced Engineering Technology And Sciences*. <https://doi.org/10.30574/wjaets.2025.15.1.0466>
- Rathaur, S., Kamath, N., & Ghanekar, U. (2020, July 1). Software Defect Density Prediction Based On Multiple Linear Regression. <https://doi.org/10.1109/icirca48905.2020.9183110>
- Richardson, C. (2023). *Microservices Patterns: With Examples In Java And Distributed Regression Testing*. Manning Publications. [https://books.google.co.id/books?id=qtgzeaaaqbaj&lpg=pt21&ots=95g24ypld8&dq=richardson%2c%20\(2023\).%20microservices%20patterns%3a%20with%20examples%20in%20java%20and%20distributed%20regression%20testing.%20manning%20publications.&lr&pg=pt21#v=onepage&q&f=false](https://books.google.co.id/books?id=qtgzeaaaqbaj&lpg=pt21&ots=95g24ypld8&dq=richardson%2c%20(2023).%20microservices%20patterns%3a%20with%20examples%20in%20java%20and%20distributed%20regression%20testing.%20manning%20publications.&lr&pg=pt21#v=onepage&q&f=false)
- Salagundi, R. M., & Savitha, R. (2024). Automated Reporting And Statistical Analysis Of Test Case Results In Continuous Integration: A Custom Dashboard Approach. *World Journal Of Advanced Engineering Technology And Sciences*. <https://doi.org/10.30574/wjaets.2024.13.1.0374>
- Sharma, S., & Chande, S. V. (2023). Optimizing Test Case Prioritization Using Machine Learning Algorithms. *Journal Of Autonomous Intelligence*. <https://doi.org/10.32629/jai.v6i2.661>
- Silva, P., & Siriwardana, A. (2023). A Comparative Study Of Qa Models Across Organizations. *Journal Of Software Engineering Research And Development*. <https://doi.org/10.21203/rs.3.rs-3458412/v1>
- Sinaga, A. M., & Sitanggang, M. (2023, September 7). Performance Comparison Of Katalon Studio And Appium On Investree Application In Funding Module. <https://doi.org/10.1109/icodse59534.2023.10291590>
- Sodhi, N. S., Singla, S., Batra, I., & Malik, A. (2025). Smart Performance Analysis Of Security-Enabled Mobile Apps For Banking Transactions. <https://doi.org/10.4018/979-8-3693-6925-8.ch008>
- Srikanth Srinivas, Er. Lagan Goel, (2025). Designing And Implementing Robust Test Automation Frameworks Using Cucumber-Bdd And Java. *Universal Research Reports*. <https://doi.org/10.36676/urr.v12.i1.1508>
- Syita Luginawati, Ari Purno Wahyu, (2023). Pengujian Perangkat Lunak Menggunakan Metode Regression Testing Pada Aplikasi Pembelajaran Matematika Untuk Siswa Tingkat Smp Berbasis Android. *Jurnal Syntax Idea*. Vol. 5, No. 1, Januari 2023. <https://doi.org/10.36418/syntax-idea.v5i1.2011>
- Test Automation. (2023). https://doi.org/10.1007/978-1-4842-9514-4_7
- Wakili, A. A., Alhassan, L. N., & Kamagata, A. (2024). Quality Assurance Practices In Agile Methodology. <https://doi.org/10.48550/arxiv.2411.05134>