

## **Perbandingan Pengujian Manual dan Terotomasi pada Software Enterprise Resource Planning**

**Rahmat Fauzan<sup>\*1)</sup>, Ferina Putri Soedjono<sup>2)</sup>, Annisa Ayu Permadani<sup>3)</sup>, dan Muhammad Ainul Yaqin<sup>4)</sup>**

<sup>1, 2, 3, 4)</sup> Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Jalan Gajayana No.50, Dinoyo, Kec. Lowokwaru, Kota Malang, 65144, Indonesia  
Email: [210605110125@student.uin-malang.ac.id](mailto:210605110125@student.uin-malang.ac.id)<sup>1)</sup>, [210605110129@student.uin-malang.ac.id](mailto:210605110129@student.uin-malang.ac.id)<sup>2)</sup>, [210605110139@student.uin-malang.ac.id](mailto:210605110139@student.uin-malang.ac.id)<sup>3)</sup>, [yaqinov@ti-uin-malang.ac.id](mailto:yaqinov@ti-uin-malang.ac.id)<sup>4)</sup>

### **Abstrak**

*Pengujian pada software ERP (Enterprise Resource Planning) merupakan aspek penting dalam pengembangan dan pemeliharaan sistem ERP. Namun, software ERP memiliki sistem yang kompleks sehingga memerlukan pengujian yang efisien dan akurat. Untuk menyelesaikan permasalahan tersebut, kami melakukan pengujian fungsional pada software ERP secara manual dan otomatis. Dari pengujian tersebut, kami membandingkan efisiensi dari pengujian manual dan otomatis pada software, khususnya pada software ERP. Dalam pengujian otomatis kami menggunakan Selenium IDE sebagai alat penguji sedangkan pengujian manual dilakukan dengan melibatkan manusia sebagai penguji. Setelah pengujian dilakukan, didapatkan hasil bahwa pengujian otomatis lebih efisien dalam segi waktu. Sedangkan dalam segi tingkat kesuksesan, pengujian manual dan otomatis memiliki hasil yang sama yakni 93,75%.*

**Kata kunci:** *Manual Testing, Automated Testing, ERP Software*

### **1. Pendahuluan (Introduction)**

ERP (Enterprise Resource Planning) adalah perangkat lunak dalam dunia bisnis yang memungkinkan perusahaan untuk mengelola seluruh proses bisnis mereka menjadi satu platform terpusat (Gupta, 2000). Software ERP biasa digunakan oleh perusahaan untuk mengontrol manajemen perusahaan sehingga sumber daya dapat meningkat (Susrama Gede dan Agustiono Wahyudi, 2022). Software ERP memiliki fungsi penting dalam mengelola data dan informasi bisnis, termasuk keuangan, persediaan, produksi, distribusi, dan lain sebagainya. Jika terjadi kesalahan pada software dapat menyebabkan kegagalan sistem dan mengganggu proses bisnis sehingga berdampak pada produktivitas dan efisiensi perusahaan. Sehingga pengujian pada software ERP sangat penting dilakukan agar sistem pada ERP dapat berjalan sesuai dengan kebutuhan bisnis (Urem ,dkk, 2014).

Software ERP dapat diuji menggunakan 2 metode pengujian, yaitu pengujian manual dan terotomasi (Rajkumar, 2023). Pengujian manual adalah pengujian yang dilakukan oleh manusia dengan mengikuti skenario pengujian yang telah ditentukan sebelumnya untuk mengevaluasi fungsionalitas suatu sistem. Pengujian manual dapat memakan waktu karena bergantung pada manusia untuk mengidentifikasi dan melaporkan masalah dalam sistem. Namun, pengujian manual berguna untuk menguji skenario yang kompleks yang tidak dapat dengan mudah diotomatisasi serta untuk mengeksplorasi pengalaman pengguna dalam menggunakan suatu sistem (Hamilton, 2020).

Pengujian terotomasi, di sisi lain, melibatkan penggunaan perangkat lunak untuk mengotomatisasi eksekusi skenario pengujian dan kasus pengujian. Pengujian terotomasi mampu menghemat waktu dan upaya yang diperlukan dalam melakukan pengujian. Selain itu, pengujian terotomasi dapat dengan mudah disesuaikan untuk menguji skenario yang kompleks dan melibatkan volume data yang besar. Namun, pengujian otomatis mungkin tidak dapat mengidentifikasi masalah yang tidak tercakup oleh skenario pengujian, dan mungkin tidak dapat mereplikasi pengalaman pengguna sistem secara akurat (Madeline, 2022).

Pada pengujian kali ini, tim penulis menguji software menggunakan metode black-box yang dilakukan secara manual dan terotomasi. Pengujian black-box adalah pengujian software yang menguji dari segi fungsionalitas suatu software (Jaya, 2018). Artinya, pengujian black-box tidak memeriksa struktur internal atau kode sumber sistem software, melainkan menguji input dan output dari software tersebut. Pengujian black-box diuji dengan menggunakan teknik pengujian fungsional yang menguji setiap fitur yang tersedia dalam software tersebut supaya berfungsi sesuai kebutuhan (Ostrand, 2002).

Pengujian black-box memiliki berbagai teknik, salah satunya adalah Boundary Value Analysis (BVA). Zuriati, dkk. (2018) menyatakan bahwa Boundary Value Analysis (BVA) adalah pengujian pada bagian input nilai yang menguji batas bawah dan batas atas dari suatu nilai inputan. BVA digunakan untuk mengidentifikasi masalah atau kesalahan pada perangkat lunak dengan menguji nilai-nilai batas atau terdekat dari batas rentang nilai input yang diizinkan. Dalam BVA, pengujian dilakukan dengan menguji nilai input pada batas-batas nilai minimum dan maksimum yang diizinkan oleh sistem, serta pada nilai di antara batas tersebut. Tujuannya adalah untuk mengidentifikasi masalah yang mungkin terjadi pada nilai batas atau saat nilai input mendekati batas rentang nilai yang diizinkan.

Dalam konteks pengujian software ERP, pengujian manual dan otomatis memiliki kelebihan dan kekurangan yang sama seperti dalam pengujian software lainnya. Namun, karena software ERP sangat penting bagi perusahaan dan dapat mempengaruhi banyak aspek bisnis, maka perbandingan antara pengujian manual dan otomatis menjadi lebih penting dan relevan untuk dilakukan (Ramlar, dkk, 2013). Software ERP memiliki kompleksitas sistem yang tinggi dan luas. Oleh karena itu, pengujian yang efisien dan akurat terhadap sistem ERP menjadi sangat penting untuk memastikan bahwa software berfungsi dengan baik dan dapat memenuhi kebutuhan bisnis yang kompleks. Dalam konteks ini, perbandingan antara pengujian manual dan terotomasi muncul sebagai pertanyaan krusial yang berkaitan dengan efisiensi pengujian.

Sebuah pengujian dapat dikatakan efisien jika hasil yang didapatkan tercapai dan sumber daya yang digunakan optimal (Mailewa, dkk, 2015). Hal ini mencakup penentuan jumlah dan jenis personel yang tepat serta peralatan dan alat pengujian yang diperlukan. Penggunaan sumber daya yang tepat membantu menghindari pemborosan dan memaksimalkan hasil pengujian. Penggunaan metode pengujian yang tepat juga dapat meningkatkan efisiensi dengan cara waktu dan upaya yang diperlukan dikurangi untuk menjalankan skenario pengujian. Dengan demikian, studi perbandingan ini dilakukan untuk menemukan metode pengujian yang paling efisien untuk software ERP sehingga dapat meminimalkan risiko kesalahan dan memaksimalkan keuntungan bagi perusahaan

## **2. Metode Penelitian (Methods)**

Penggunaan metode black-box testing ini mengaplikasikan teknik Boundary Value Analysis (BVA) untuk membandingkan pengujian software secara manual dan otomatis. Adapun tahapan penelitian ini berupa, identifikasi masalah, pemilihan metode yang digunakan, dan pemilihan modul yang akan diuji, kemudian melakukan proses pengujian software, menghitung data testing dan dilanjutkan dengan menganalisis perbandingan dari data testing pengujian manual dengan pengujian otomatis.

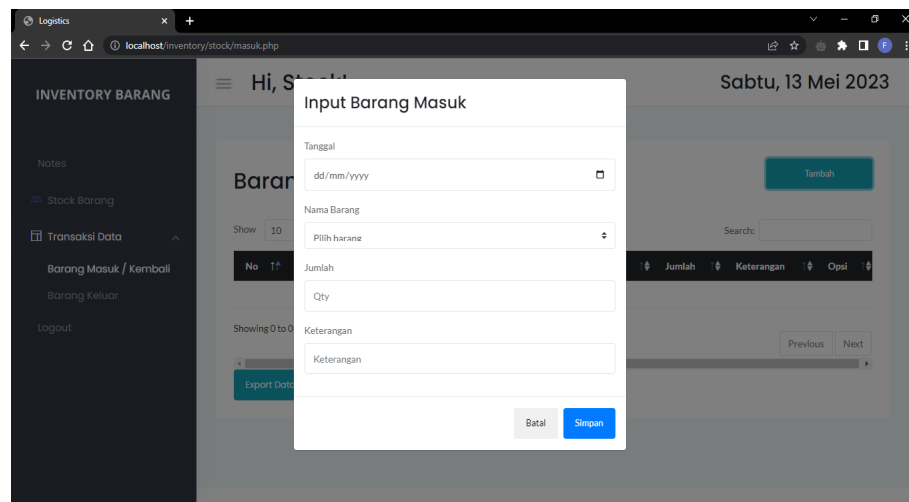
Pengujian manual dilakukan dengan manusia sebagai penguji sementara pengujian terotomasi menggunakan alat bantu berupa software. Software yang digunakan dalam studi perbandingan ini adalah Selenium IDE, sebuah alat yang digunakan untuk melakukan pengujian terotomasi pada aplikasi web. Dalam pengujian otomatis, Selenium dapat membantu mempercepat proses pengujian otomatis.

Berikut merupakan langkah-langkah untuk melakukan pengujian perangkat lunak menggunakan Selenium IDE :

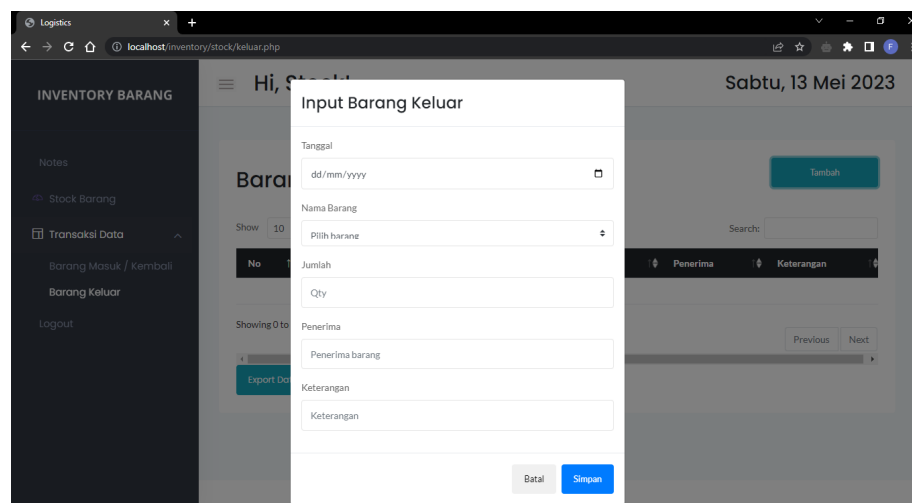
1. Menginstall dan mengaktifkan Selenium IDE pada browser Chrome  
Selenium IDE adalah plugin yang tersedia dalam beberapa aplikasi browser salah satunya Chrome. Unduh dan install Selenium IDE dari website resminya kemudian aktifkan plugin tersebut pada browser.
2. Merekam aksi penggunaan aplikasi web yang akan diuji  
Buka aplikasi web yang akan diuji pada browser Chrome dan aktifkan Selenium IDE. Kemudian, klik tombol "Record a new test" pada Selenium IDE untuk memulai merekam aksi pengguna. Lalu lakukan beberapa aksi pengguna pada aplikasi web, seperti mengisi form, menekan tombol, atau mengakses halaman lain.
3. Menganalisis script pengujian yang dihasilkan  
Setelah merekam aksi pengguna, Selenium IDE akan menghasilkan script pengujian dalam format HTML atau beberapa format lainnya. Analisis script tersebut untuk memastikan bahwa semua aksi pengguna sudah direkam dengan benar.
4. Menjalankan script pengujian  
Setelah memastikan script pengujian telah benar, kemudian jalankan script tersebut pada browser Chrome. Selenium IDE akan memutar kembali semua aksi pengguna yang telah direkam dan menghasilkan laporan pengujian.
5. Menganalisis hasil pengujian  
Setelah script pengujian selesai dijalankan, penguji menganalisis laporan pengujian yang dihasilkan oleh Selenium IDE. Jika terdapat kesalahan atau bug, maka penguji memperbaiki script pengujian dan menjalankan kembali sampai hasilnya sesuai dengan yang diharapkan.

Kami memilih metode black-box dalam pengujian manual dan otomatis pada software inventory. Metode ini dipilih karena berfokus pada pengujian fungsionalitas dan kinerja sistem secara keseluruhan tanpa memperhatikan rincian atau keadaan internal dari sistem tersebut. Dalam konteks ERP, pengujian black-box menjadi penting karena perangkat lunak ini memiliki kompleksitas yang tinggi dan seringkali digunakan untuk menangani proses bisnis yang krusial bagi perusahaan.

Penelitian ini akan berfokus pada pengujian manual dan otomatis pada bagian inventori dari perangkat lunak ERP. Inventori merupakan salah satu bagian penting dalam manajemen bisnis yang berkaitan dengan pengelolaan stok barang dan pengiriman produk. Dalam inventori pengguna dapat melihat dan menambah stok barang, mendata barang keluar, dan mendata barang masuk. Modul yang diuji pada bagian inventori ini berupa halaman barang masuk dan halaman barang keluar. Inventori perlu diuji untuk memastikan keakuratan dan keandalan fitur secara fungsional dalam proses pengumpulan data. Dengan melakukan pengujian software, perusahaan dapat memastikan bahwa data yang disimpan dalam software aman dan akurat, sehingga dapat membantu perusahaan dalam pengelolaan stok barang yang lebih baik.



Gambar 1. Tampilan Barang Masuk



Gambar 2. Tampilan Barang Keluar

Dalam melakukan pengujian, penting untuk membuat kasus uji atau test case terlebih dahulu. Test case adalah skenario yang digunakan untuk menguji perangkat lunak atau sistem. Test case terdiri dari input, aksi atau kejadian dan respon yang diharapkan. Tujuan penulisan test case terutama untuk memvalidasi cakupan uji aplikasi.

Struktur test case terdiri dari :

1. ID Test Case : Nomor identifikasi test case.
2. Sample Data : Deskripsi test case.
3. Expected Result : Hasil yang diharapkan untuk keluar setelah prosedur test case dilakukan.
4. Result : Hasil aktual yang keluar setelah prosedur test case dilakukan.
5. Conclusion : Menampilkan status test case apakah berhasil atau gagal berdasarkan perbandingan hasil aktual dan hasil yang diharapkan.

Setelah hasil uji perangkat lunak tercatat dalam tabel test case, langkah selanjutnya dilakukan perhitungan akurasi pengujian guna membandingkan tingkat akurasi di antara pengujian manual dan terotomasi. Dalam perhitungan nilai akurasi suatu pengujian, nilai akurasi diperoleh dengan menghitung persentase dari jumlah pengujian yang berhasil sukses dibagi dengan total test case yang diuji. Rumus untuk menghitung nilai akurasi pengujian adalah sebagai berikut:

$$Akurasi = \frac{Jumlah\ Pengujian\ Sukses}{Total\ Test\ Case} \times 100\ %$$

Rumus ini memungkinkan untuk mengukur sejauh mana pengujian berhasil dalam menghasilkan hasil yang benar dalam konteks kasus uji yang diberikan. Dengan menghitung nilai akurasi, kita dapat mengukur kinerja suatu sistem atau model dalam mengatasi berbagai kasus uji yang berbeda.

### 3. Hasil dan Pembahasan (Results and Discussions)

Pada test case yang telah dibuat, dilakukan pengujian pada modul transaksi barang yang terbagi menjadi Barang Masuk dan Barang Keluar. Pada Barang masuk memiliki input berupa tanggal, nama barang, jumlah, dan keterangan. Sedangkan Barang Keluar memiliki input berupa tanggal, nama barang, jumlah, penerima, dan keterangan.

Tabel 1 adalah tabel hasil percobaan pada form Barang Masuk dengan 12 pengujian. Dari 12 pengujian tersebut, tingkat kesuksesan pengujian manual dan otomatis adalah 100%. Pengujian manual memerlukan waktu sebanyak 161,97 detik dan pengujian otomatis memerlukan waktu sebanyak 83,74 detik untuk melaksanakan pengujian.

**Tabel 1.** Hasil Test Case Barang Masuk / Kembali

ID	Sample Data	Expected Result	Result		Conclusion	
			Manual	Otomatis	Manual	Otomatis
<b>Tanggal</b>						
TC1	Mengosongkan tanggal	F	F	F	Success	Success
TC2	Menginput tanggal melebihi jumlah tanggal dalam bulan	F	F	F	Success	Success
TC3	Menginput tanggal	T	T	T	Success	Success
<b>Nama Barang</b>						
TC4	Mengosongkan nama barang	F	F	F	Success	Success
TC5	Menginput nama barang yang tidak ada di pilihan	F	F	F	Success	Success
TC6	Menginput nama barang sesuai yang tersedia di pilihan	T	T	T	Success	Success
<b>Jumlah</b>						
TC7	Menginput jumlah dengan panjang karakter 0	F	F	F	Success	Success
TC8	Menginput jumlah dengan panjang karakter 1	T	T	T	Success	Success
TC9	Menginput jumlah dengan panjang karakter 11	T	T	T	Success	Success
TC10	Menginput jumlah dengan panjang karakter 12	F	F	F	Success	Success
<b>Keterangan</b>						
TC11	Mengosongkan keterangan	T	T	T	Success	Success
TC12	Menginput keterangan	T	T	T	Success	Success

**Tabel 2.** Hasil Test Case Barang Keluar

ID	Sample Data	Expected Result	Result		Conclusion	
			Manual	Otomatis	Manual	Otomatis
<b>Tanggal</b>						
TC1	Mengosongkan tanggal	F	F	F	Success	Success
TC2	Menginput tanggal melebihi jumlah tanggal dalam bulan	F	F	F	Success	Success
TC3	Menginput tanggal	T	T	T	Success	Success
<b>Nama Barang</b>						
TC4	Mengosongkan nama barang	F	F	F	Success	Success
TC5	Menginput nama barang yang tidak ada di pilihan	F	F	F	Success	Success
TC6	Menginput nama barang sesuai yang tersedia di pilihan	T	T	T	Success	Success

<b>Jumlah</b>						
TC7	Menginput jumlah dengan panjang karakter 0	F	F	F	Success	Success
TC8	Menginput jumlah dengan panjang karakter 1	T	T	T	Success	Success
TC9	Menginput jumlah sama dengan stock	T	T	T	Success	Success
TC10	Menginput jumlah lebih besar dari stock	F	T	T	Failed	Failed
<b>Penerima</b>						
TC11	Menginput penerima dengan panjang karakter 0	F	T	T	Failed	Failed
TC12	Menginput penerima dengan panjang karakter 1	T	T	T	Success	Success
TC13	Menginput penerima dengan panjang karakter 35	T	T	T	Success	Success
TC14	Menginput penerima dengan panjang karakter 36	F	F	F	Success	Success
<b>Keterangan</b>						
TC15	Mengosongkan keterangan	T	T	T	Success	Success
TC16	Menginput keterangan	T	T	T	Success	Success

Tabel 2 adalah tabel hasil percobaan pada form Barang Keluar dengan 16 pengujian. Dari 16 pengujian tersebut, terdapat 2 pengujian yang gagal pada pengujian manual dan otomatis sehingga tingkat kesuksesan pada pengujian manual dan otomatis pada form Barang Keluar adalah sebesar 87,5%. Pengujian manual memerlukan waktu sebanyak 246,88 detik dan pengujian otomatis memerlukan waktu sebanyak 100,99 detik untuk melaksanakan pengujian.

**Tabel 3.** Hasil Perbandingan Akurasi Pengujian

No	Modul	Akurasi (%)	
		Manual	Otomatis
1	Barang Masuk	100%	100%
2	Barang Keluar	87,5%	87,5%
<b>Rata-rata</b>		93,75%	93,75%

Tabel 3 adalah hasil akurasi kesuksesan pada pengujian manual dan otomatis. Akurasi kesuksesan pada pengujian manual dan otomatis sama yaitu 100% dan 87,5%. Rata-rata akurasi dari pengujian ini adalah 93,75%.

**Tabel 4.** Hasil Perbandingan Waktu Pengujian

No	Modul	Waktu (s)	
		Manual	Otomatis
<b>Barang Masuk</b>			
1	Tanggal	45.69 detik	28.64 detik
2	Nama Barang	36.62 detik	20.84 detik
3	Jumlah	59.99 detik	16.52 detik
4	Keterangan	19.67 detik	17.74 detik
<b>Total</b>		161,97 detik	83,74 detik
<b>Barang Keluar</b>			
1	Tanggal	33.33 detik	25.32 detik
2	Nama Barang	41.65 detik	19.62 detik
3	Jumlah	70.54 detik	15.44 detik
4	Penerima	76.21 detik	16.26 detik
5	Keterangan	25.15 detik	24.35 detik
<b>Total</b>		246,88 detik	100,99 detik

Tabel 4 adalah hasil waktu pengujian pada pengujian manual dan otomatis. Dalam segi waktu, pengujian otomatis memiliki hasil yang lebih cepat dari pada pengujian manual. Total waktu pengujian manual pada barang masuk adalah 161,97 detik dan pengujian otomatisnya 82,74 detik. Sedangkan

waktu pengujian manual pada barang keluar adalah 246,88 detik dan pengujian otomatisnya adalah 100,99 detik

#### **4. Kesimpulan (Conclusion)**

Setelah dilakukannya penelitian, maka bisa disimpulkan bahwa pengujian otomatis merupakan metode yang paling efisien untuk pengujian fungsionalitas. Melalui tabel hasil perbandingan metode pengujian, didapat bahwa lama waktu pengujian otomatis sekitar 2-3 kali lebih cepat dari pada pengujian manual. Hal ini membuktikan bahwa pengujian otomatis lebih efisien dalam segi waktu. Efisiensi waktu ini dapat terjadi karena pengujian otomatis menggantikan keterlibatan manusia dengan alat pengujian. Selain itu, pengujian terotomasi memberikan konsistensi dan akurasi yang tinggi. Alat pengujian otomatis dapat menjalankan serangkaian langkah pengujian dengan konsisten setiap kali dieksekusi. Hal ini membantu dalam mendeteksi masalah dan bug, serta meminimalisir kesalahan manusia yang mungkin terjadi dalam pengujian manual.

Meskipun pengujian terotomasi memiliki keuntungan dalam pengujian fungsional, perlu diperhatikan bahwa tidak semua aspek pengujian dapat dilakukan secara otomatis. Beberapa aspek pengujian, seperti pengujian User Interface (UI) dan pengujian pengalaman pengguna, lebih tepat diuji dengan pengujian manual. Hal ini dikarenakan pengujian UI dan pengalaman pengguna melibatkan aspek-aspek seperti desain, kenyamanan, dan interaksi yang sulit untuk diukur atau dinilai jika dilakukan secara otomatis. Oleh karena itu, manusia merupakan penguji yang lebih baik dalam menafsirkan dan mengevaluasi pengalaman pengguna dan desain.

#### **Ucapan Terima Kasih (Acknowledgement)**

Penelitian yang tim penulis lakukan telah mendapat berbagai bantuan dari banyak pihak. Oleh karena itu, tim penulis mengucapkan terima kasih kepada dosen pengampu yang telah memberikan pengarahan dalam berlangsungnya penelitian ini.

#### **Daftar Pustaka**

- Benefits of Automated Testing Over Manual Testing. [online]. <https://www.cprime.com/resources/blog/benefits-of-automated-testing-over-manual-testing/> (Diakses 13 April 2023).
- Gupta, A., 2000. Enterprise resource planning: the emerging organizational value systems. *Ind. Manag. Data Syst.* 100, 114–118. (Opatija, 24 Juli 2014).
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol. 3, No. 2, hal. 45-48.
- Mailewa, A., Herath, J., Herath, S., 2015. A survey of effective and efficient software testing, in: *The Midwest Instruction and Computing Symposium (MICS)*, Grand Forks, ND.
- Manual Testing Tutorial: What is, Types, Concepts. [online]. <https://www.guru99.com/manual-testing.html> (Diakses 13 April 2023).
- Manual Testing Vs Automation Testing In Software Testing. [online]. <https://www.softwaretestingmaterial.com/manual-testing-vs-automation-testing/> (Diakses 13 April 2023).
- Ostrand, T., 2002. Black-Box Testing. *Encycl. Softw. Eng.*
- Ramler, R., Kopetzky, T., dan Platz, W. (2013). A Business View on Testing ERP Systems with Value-Based Requirements Coverage. *Lecture Notes in Information Systems and Organization, in: Felix Piazzolo & Michael Felderer (ed.), Innovation and Future of Enterprise Information Systems*, edition 127, hal. 219-234, Springer.
- Susrama Gede, Agustiono Wahyudi, 2022. *Buku Ajar Enterprise Resource Planning (ERP)*.

- Urem, F., Fertalj, K., Livaja, I., 2014. Modeling of software failure cost in ERP systems, in: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, pp. 557–561.
- Zuriati, Widyawati, D. K., Sitanggang, I. S., dan Buowo, A. (2018). Teknik Pengujian Boundary Value Analysis pada Aplikasi Learning Management System Polinela. *Jurnal TAM (Technology Acceptance Model)*, Vol. 9, No. 2, hal. 86-92.