



## Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket

Rahman Abdillah<sup>1\*</sup>, Rudi Hermawan<sup>2</sup>, Wawan Hermawansyah<sup>3</sup>, Ibnu Adkha<sup>4</sup>,  
Heri Arifin<sup>5</sup>

<sup>1,2</sup> Universitas Indraprasta PGRI, Indonesia

<sup>3</sup> Universitas Bani Saleh, Indonesia

<sup>4,5</sup> Politeknik Bhakti Kartini, Indonesia

[rabdil.bu@gmail.com](mailto:rabdil.bu@gmail.com)<sup>1\*</sup>

Alamat: Jl. Nangka Raya No.58C, Tanjung Barat, Kec.Jagakarsa – Jakarta Selatan

Korespondensi penulis: [rabdil.bu@gmail.com](mailto:rabdil.bu@gmail.com)

**Abstract.** *Software testing is one of the aspects in software design methodology. The author conducted research by taking a case study on the inventory stock information system in a goods delivery service business. The testing technique for this information system uses the black box testing method, with the Equivalence Partitioning technique. From the testing results, the author found irregularities in the database and conducted a more in-depth analysis of the class diagram. The diversity of software types suggests the importance of flexibility and improvisation during software testing, such as determining the appropriate test cases, selecting the appropriate testing methods, and determining the suitable testing environment. Software testing should also consider the effectiveness of resources, project scheduling and deadlines, user/client requests, the availability of testing infrastructure, and other parameters that may be more complex.*

**Keywords:** *black box testing, class diagram, equivalence partitioning, inventory information system.*

**Abstrak.** Pengujian perangkat lunak merupakan salah satu aspek yang terdapat dalam metode perancangan perangkat lunak. Penulis melakukan penelitian dengan mengambil studi kasus pada sistem informasi inventori stok barang pada suatu usaha jasa pengiriman barang. Teknik pengujian sistem informasi ini menggunakan metode *black box testing*, dengan teknik pengujian *Equivalence Partitioning*. Dari hasil pengujian, penulis menemukan kejanggalan pada database dan melakukan analisis yang lebih mendalam pada bagian *class diagram*. Keragaman jenis perangkat lunak mengisyaratkan pentingnya fleksibilitas dan improvisasi saat pengujian perangkat lunak dilakukan, seperti penentuan uji kasus yang tepat, pemilihan metode pengujian yang sesuai, penentuan lingkungan pengujian yang sesuai. Pengujian perangkat lunak seharusnya juga mempertimbangkan efektifitas sumber daya, penjadwalan dan tenggat waktu proyek, permintaan user/client, ketersediaan infrastruktur pengujian serta parameter-parameter lainnya yang mungkin saja lebih kompleks.

**Kata kunci:** *black box testing, class diagram, equivalence partitioning, sistem informasi inventori.*

### 1. LATAR BELAKANG

Dalam era digital saat ini, perangkat lunak (*software*) telah menjadi aspek penting dalam kehidupan kita sehari-hari, mulai dari bidang bisnis, manajemen, pendidikan, kesehatan, hingga hiburan. Pertumbuhan pesat teknologi informasi dan komunikasi telah meningkatkan kompleksitas sistem perangkat lunak yang kita gunakan sehari-hari. Dalam proses perancangan dan pengembangan perangkat lunak, setiap kesalahan atau bug dalam perangkat lunak dapat menyebabkan gangguan yang cukup signifikan, baik dari segi operasional maupun keamanan. Hal ini menekankan pentingnya tahap pengujian perangkat lunak yang efektif dan efisien. Selanjutnya, pengujian perangkat lunak telah berkembang menjadi disiplin yang kompleks

dengan berbagai metode dan teknik untuk mendeteksi bug, mengurangi risiko, dan memastikan bahwa perangkat lunak telah memenuhi persyaratan pengguna.

Dalam teori rekayasa perangkat lunak, salah satu metode yang sering digunakan untuk pengembangan perangkat lunak adalah Waterfall model atau Siklus Hidup Pengembangan Sistem Perangkat Lunak/SDLC (System Development Life Cycle). Dalam kaitannya dengan metode pengembangan SDLC, perancangan perangkat lunak dilakukan mulai dari tahap perencanaan (planning), analisis (analysis), desain (design), implementasi (implementation) (Sommerville, 2012). Beberapa peneliti bidang rekayasa perangkat lunak lalu menambahkan tahap-tahap tersebut dengan proses evaluasi (evaluation), pengujian (testing) dan perawatan (maintenance) (Abdillah, 2018). Dari beberapa tahapan proses pengembangan perangkat lunak tersebut, pada karya ilmiah ini penulis membahas hal yang spesifik pada tahap pengujian (testing) dari sistem perangkat lunak.

Pengujian perangkat lunak melibatkan evaluasi sistem atau komponennya dengan tujuan untuk menemukan kesalahan dan memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang ditentukan. Proses ini melibatkan berbagai tahapan, mulai dari pengujian unit, pengujian integrasi, hingga pengujian sistem dan penerimaan (acceptability) dari pengguna (Pressman, 2015). Setiap tahap memiliki tujuan dan metodologi yang berbeda untuk menangani aspek-aspek yang spesifik dari perangkat lunak. Selain itu, pengujian perangkat lunak juga mencakup teknik pengujian secara manual dan otomatis, yang keduanya memiliki peran penting dalam siklus hidup pengembangan perangkat lunak. Dengan meningkatnya kompleksitas sistem perangkat lunak modern, pendekatan pengujian yang efektif dan efisien menjadi sangat penting untuk menjamin kualitas dan keandalan produk akhir.

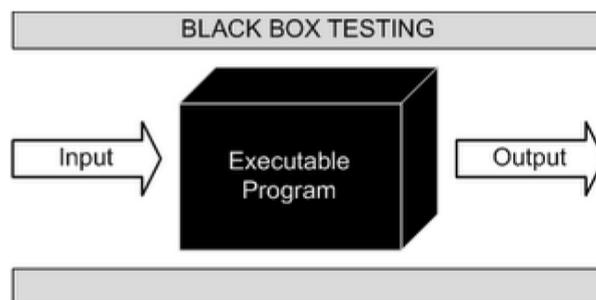
Pengujian perangkat lunak adalah proses krusial dalam pengembangan perangkat lunak yang bertujuan untuk memastikan kualitas, kinerja, dan keamanan dari produk yang dikembangkan. Dalam beberapa dekade terakhir, perkembangan teknologi informasi dan komunikasi telah meningkatkan kebutuhan akan perangkat lunak yang andal dan efisien. Pengujian perangkat lunak tidak hanya berfokus pada menemukan kesalahan, tetapi juga bertujuan untuk meningkatkan kualitas perangkat lunak secara keseluruhan. Hal ini termasuk meningkatkan pengalaman pengguna (user), memastikan bahwa sistem dapat menangani beban kerja yang diharapkan, serta memverifikasi bahwa semua fungsi perangkat lunak dapat bekerja dengan benar dalam berbagai kondisi. Dengan demikian, pengujian perangkat lunak menjadi bagian yang tidak dapat dipisahkan dari tahapan pengembangan perangkat lunak, serta dapat

meyakinkan pengguna (user) bahwa perangkat lunak tersebut telah memenuhi standar kualitas (Wardhani & Abdillah, 2018).

## 2. KAJIAN TEORITIS

Pengujian perangkat lunak adalah tahap penting dalam pengembangan perangkat lunak dikarenakan pengujian ini berfungsi untuk memastikan kualitas yang baik dari segi perancangan dan struktur kontrol pemrogramannya (Setiawan, 2017). Pengujian ini juga merupakan aspek integral dari pengembangan sistem perangkat lunak, yang berfungsi sebagai proses validasi dan verifikasi dari perangkat lunak tersebut. Selain itu, pengujian perangkat lunak adalah elemen penting dalam jaminan standar kualitas perangkat lunak yang merupakan bagian tak terpisahkan dari siklus hidup pengembangan perangkat lunak (Sommerville, 2012).

Tahap pengujian perangkat lunak memerlukan biaya yang cukup besar, bahkan bisa dibilang mahal. Berdasarkan tinjauan dari beberapa literatur tentang pengujian perangkat lunak, sekitar 50% dari total biaya pengembangan perangkat lunak digunakan untuk tahap pengujian (Jorgensen, 2014). Untuk mengurangi biaya pengujian yang cukup besar, pengembangan pengujian perangkat lunak secara otomatis dapat dilakukan sebagai solusi. Terdapat beberapa teknik yang dapat diterapkan dalam pengujian perangkat lunak, antara lain metode blackbox dan whitebox. Metode blackbox berfokus pada pengujian fungsionalitas perangkat lunak, sedangkan metode whitebox menguji struktur program. Setiap metode memiliki kekurangan dan kelebihan masing-masing. Misalnya, metode blackbox, yang berfokus pada fungsionalitas dan masukan (input), bisa mengalami kesulitan dalam membuat dokumentasi hasil pengujian jika spesifikasi perangkat lunak tidak jelas. Sementara itu, metode whitebox, yang menitikberatkan pada struktur program, membutuhkan banyak sumber daya ketika diterapkan pada perangkat lunak berskala besar. Metode ini juga memerlukan penguji yang memahami kode program yang sedang diuji. Meskipun pengujian struktural memegang peranan penting, terutama karena lebih efektif dalam mengurangi biaya pengujian perangkat lunak dibandingkan dengan pengujian fungsional (Sulistyanto, 2014).



**Gambar 1.** Teknik Pengujian Perangkat Lunak Black Box Testing

Pengujian dilakukan dengan cara mengevaluasi konfigurasi perangkat lunak, yang meliputi spesifikasi kebutuhan, deskripsi perancangan, dan program yang dihasilkan (Wahyu & Afrizal, 2023). Hasil evaluasi ini kemudian dibandingkan dengan hasil yang diharapkan dari pengujian. Jika terdapat kesalahan, perangkat lunak harus diperbaiki dan diuji kembali. Oleh karena itu, pengujian pada dasarnya dapat dianggap lebih sebagai aktivitas perusakan daripada pembangunan. Namun demikian, pentingnya pengujian perangkat lunak dan dampaknya terhadap kualitas perangkat lunak tidak dapat diabaikan, karena melibatkan serangkaian aktivitas produksi di mana kemungkinan kesalahan manusia sangat tinggi. Oleh karena itu, pengembangan perangkat lunak seharusnya selalu disertai dengan aktivitas penjaminan kualitas. Berikut ini merupakan beberapa metode pengujian perangkat lunak (Wibisono & Baskoro, 2002)

### **Pengujian Unit (*Unit Testing*)**

Pengujian unit adalah proses menguji bagian terkecil dari perangkat lunak yang dapat diuji, seperti fungsi, prosedur, atau metode, secara terpisah. Tujuan dari pengujian ini adalah untuk memverifikasi bahwa setiap unit berfungsi sesuai dengan desainnya. Pengujian unit biasanya dilakukan oleh pengembang perangkat lunak selama fase pengembangan dan sering kali diotomatisasi untuk meningkatkan efisiensi.

### **Pengujian Integrasi (*Integration Testing*)**

Pengujian integrasi dilakukan untuk menguji kombinasi dari dua atau lebih unit perangkat lunak yang telah diuji secara individual. Tujuannya adalah untuk mendeteksi masalah antarmuka antara unit-unit tersebut dan memastikan bahwa mereka bekerja bersama secara harmonis. Metode ini membantu mengidentifikasi masalah yang mungkin tidak muncul saat unit diuji secara terpisah.

### **Pengujian Sistem (*System Testing*)**

Pengujian sistem adalah proses menguji seluruh sistem perangkat lunak sebagai satu kesatuan. Pengujian ini dilakukan setelah semua komponen diintegrasikan dan bertujuan untuk memastikan bahwa sistem memenuhi persyaratan fungsional dan non-fungsional. Pengujian sistem mencakup berbagai jenis pengujian, termasuk pengujian fungsional, pengujian kinerja, dan pengujian keamanan.

### **Pengujian Penerimaan (*Acceptance Testing*)**

Pengujian penerimaan adalah tahap akhir dari proses pengujian, yang dilakukan untuk memastikan bahwa perangkat lunak siap digunakan oleh pengguna akhir. Pengujian ini biasanya melibatkan pengguna atau perwakilan mereka dan bertujuan untuk memverifikasi bahwa sistem memenuhi kebutuhan dan harapan pengguna. Pengujian penerimaan dapat mencakup pengujian alfa dan beta, di mana perangkat lunak diuji dalam lingkungan pengembang dan pengguna, masing-masing.

### **Pengujian Regresi (*Regression Testing*)**

Pengujian regresi dilakukan setelah perubahan atau pembaruan dilakukan pada perangkat lunak, untuk memastikan bahwa perubahan tersebut tidak mempengaruhi fungsi yang ada. Pengujian ini penting untuk meminimalkan risiko pengenalan bug baru atau mengganggu fungsi yang sudah ada setelah pemeliharaan atau pengembangan lebih lanjut.

### **Pengujian Otomatis (*Automated Testing*)**

Pengujian otomatis menggunakan alat dan skrip untuk secara otomatis menjalankan pengujian pada perangkat lunak. Metode ini meningkatkan efisiensi dan konsistensi pengujian, terutama dalam pengujian regresi atau unit yang sering diulang. Pengujian otomatis sangat berguna dalam proyek-proyek besar yang memerlukan pengujian berulang dengan cepat dan akurat.

## **3. METODE PENGUJIAN PERANGKAT LUNAK**

Metode *Black Box Testing* adalah pengujian yang bertujuan untuk menemukan kesalahan dalam perancangan perangkat lunak atau sistem informasi, seperti jika terdapatnya kesalahan fungsi dan menu atau fitur yang tidak dapat bekerja, dan lain sebagainya (Uminingsih, 2022). *Black Box Testing* lebih tepat digunakan untuk menguji fungsionalitas dari perangkat lunak. Dalam pengujian ini, digunakan data masukan secara acak untuk menguji hasil atau keluaran dari sistem. Suatu masukan (*input*) dari pengguna jika terdapat kesalahan, data tersebut akan ditolak oleh sistem atau tidak dapat disimpan dalam basis data, sedangkan jika data masukan sudah benar, maka akan diterima atau disimpan dalam basis data sistem informasi.

Ada beberapa teknik pengujian dalam *Black Box Testing*, yaitu: a) Teknik *Equivalence Partitioning*, yaitu membagi input data menjadi beberapa partisi. b) Teknik *Boundary Value*

*Analysis*, yaitu mencari kesalahan dari sisi luar atau dalam perangkat lunak, serta nilai minimum dan maksimum dari kesalahan yang ditemukan. c) Teknik *Fuzzing*, yaitu mencari bug atau gangguan pada perangkat lunak dengan menyuntikkan data cacat. d) Teknik *Cause-Effect Graph*, yaitu menggunakan grafik untuk menggambarkan hubungan antara sebab dan akibat. e) Teknik *Orthogonal Array Testing*, teknik pengujian ini digunakan jika domain input relatif kecil, namun cukup kompleks untuk skala besar. f) Teknik *All Pair Testing*, yaitu teknik pengujian yang mendesain semua pasangan kasus pengujian (*test case*) untuk menguji semua kombinasi diskrit dari pasangan input parameter. Adapun tujuan dari teknik pengujian ini adalah memastikan semua pasangan tersebut teruji. g) Teknik *State Transition*, berguna untuk menguji kondisi mesin dan navigasi dalam bentuk grafik (Beizer, 1990).

Dalam penelitian ini, teknik pengujian yang digunakan pada Sistem Informasi Inventori Barang adalah Teknik *Equivalence Partitioning*. Teknik ini membagi data masukan dari unit perangkat lunak menjadi beberapa partisi, dari mana *test case* dapat diturunkan. Prinsip dasarnya adalah merancang kasus uji yang mencakup setiap bagian (*unit*) minimal sekali. Teknik ini berusaha mendefinisikan kasus uji yang mengungkapkan kelas kesalahan, sehingga mengurangi jumlah kasus uji yang perlu dikembangkan (Jaya, et.al. 2019).

#### 4. HASIL DAN PEMBAHASAN

Pada karya ilmiah ini penulis menggunakan teknik pengujian dengan metode *black box testing* dengan teknik pengujian *Equivalence Partitioning*. Dibawah ini merupakan contoh sistem informasi inventaris barang di gedung ekspedisi sebuah perusahaan pengiriman barang. Sistem informasi ini masih bersifat prototipe dan belum digunakan dalam perusahaan tersebut. Dari gambar *dashboard* dibawah ini, dapat kita lihat beberapa fitur atau menu layanan sistem informasi, seperti data master, data transaksi, dan menu laporan.

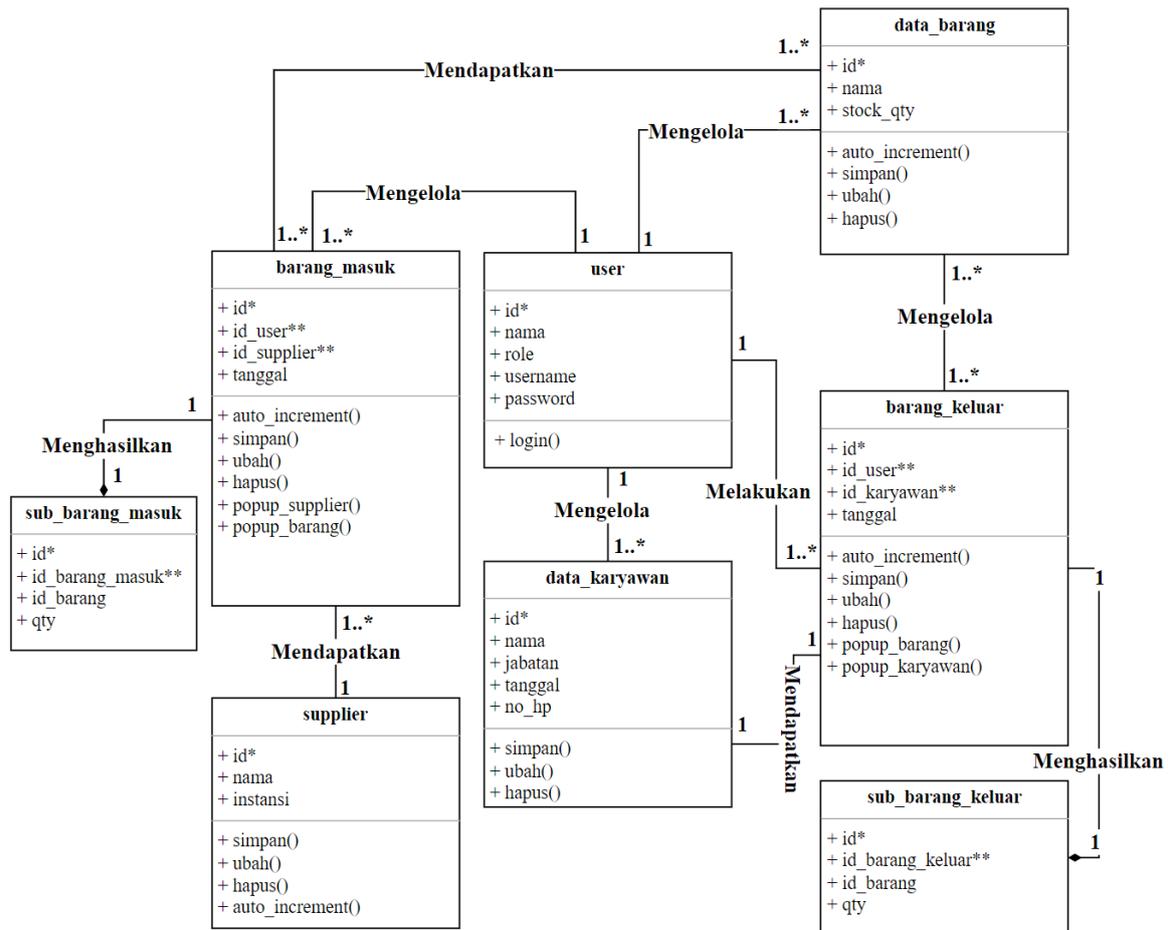


**Gambar 2.** Dashboard Sistem Informasi Inventaris Barang

Penulis melakukan pengujian terhadap fitur-fitur yang terdapat pada sistem informasi tersebut secara acak. Selanjutnya, menu data barang diuji dengan memasukkan (*input*) data-data seperti: id barang, nama barang, jenis barang dan berat. Namun, setelah diteliti lebih lanjut, tidak ada kolom untuk memasukkan stok, namun dalam form tampilan di layar sudah terdapat stok barang, sehingga pengguna (*user*) sebenarnya tidak dapat memasukkan jumlah stok melalui form data barang. Hal ini bisa terjadi karena programmer, memasukkan data stok barang langsung melalui database sql, yang seharusnya hal ini tidak terjadi.

Kode Barang	Nama	Jenis	Berat	Stok
A005	Sendal	Pakaian	500 Gram	33
BRG01	Spidol	ATK	50 Gram	132
BRG02	Pulpen	ATK	10 Gram	13
SL01	Solasi	ATK	100 Gram	200

**Gambar 3.** Form Pengisian Data Barang



**Gambar 4.** Class Diagram Sistem Informasi Inventaris Barang

Langkah berikutnya adalah penulis melakukan analisis terhadap desain database *class diagram* sistem informasi inventaris barang sehingga didapati gambar diagram seperti gambar 4 diatas. Pada tabel `data_barang` didapati entitas `stock_qty` yang berarti stok barang pada data barang telah dideskripsikan pada database sistem. Selanjutnya, dari hasil analisis tersebut diharapkan kepada pengembang (*programmer*) untuk dapat memperbaiki dengan cara menambah kolom pengisian stok barang pada form data barang. Dalam karya ilmiah ini, tidak semua teknik pengujian diimplementasikan. Penulis hanya melakukan teknik pengujian secara *Equivalence Partitioning* yang berarti hanya perbagian fitur yang diuji secara random dan perlu diperbaiki atau dikembangkan lebih lanjut oleh pengembang (*programmer*).

## 5. KESIMPULAN DAN SARAN

Pengujian perangkat lunak bertujuan untuk menjamin kualitas produk perangkat lunak yang dihasilkan. Melalui serangkaian metode pengujian yang komprehensif, perusahaan atau pengguna (*user*) dapat memastikan bahwa sistem informasi inventori barang siap digunakan dengan baik dan tepat guna. Secara garis besar, terdapat dua kategori pengujian perangkat lunak, yaitu: *white box testing* dan *black box testing*. Beberapa metode pengujian perangkat lunak terdapat kelebihan dan kekurangan dalam penggunaannya. Untuk itu, penulis menyarankan penggunaan metode pengujian perangkat lunak sebaiknya dilakukan dengan mempertimbangkan efektifitas sumber daya, tenggat waktu, permintaan user/client, ketersediaan infrastruktur dan teknologi, serta parameter lainnya yang lebih kompleks. Tahap pengujian perangkat lunak ini merupakan tahapan pengembangan perangkat lunak yang cukup penting untuk menghindari kesalahan dalam pengelolaan sistem informasi stok barang yang dapat mengakibatkan kerugian finansial atau gangguan operasional. Pengujian yang baik juga dapat meningkatkan kepercayaan pengguna (*user*) terhadap sistem, karena mereka dapat mengandalkan akurasi data yang disediakan oleh sistem untuk membuat keputusan bisnis dengan tepat. Dengan demikian, investasi dalam pengujian perangkat lunak merupakan langkah yang strategis untuk mendukung keberhasilan implementasi sistem informasi inventori barang.

## DAFTAR REFERENSI

- Arfinda, D., Selo, Lukito. (2019). Studi Literatur Pengujian Perangkat Lunak. *Proceeding SINTAK*. 15-21
- Boris Beizer, (1990). *Software Testing Techniques 2nd Edition*. Boston: International Thompson Computer Press.
- Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 131-136. <https://doi.org/10.32493/informatika.v4i4.3834>
- Jorgensen, P. C. (2014). *Software Testing 4<sup>th</sup> Edition*. Florida: Taylor & Francis Group.
- Muslimin, D. B., Kusmanto, D., Amelia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-26. <https://doi.org/10.32493/informatika.v5i1.3778>
- Prebianto, A., Abdillah, R., Valentino, V. H. (2020). Perancangan Sistem Informasi Penjualan Donat pada PT.King Alianz Donuts. *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, 01(03), 424-429. <https://doi.org/10.30998/jrami.v1i03.324>.

- Pressman, Roger. (2015). *Software Engineering: A Practitioner Approach 8<sup>th</sup> Edition*. New York : McGraw Hill.
- Setiawan, R. (2017). Pengujian Perangkat Lunak. *Seminar Nasional Informatika dan Aplikasinya (SNIA) 2017*. (c37-c39).  
[http://repository.unjani.ac.id/index.php?p=show\\_detail&id=561&keywords=](http://repository.unjani.ac.id/index.php?p=show_detail&id=561&keywords=)
- Sommerville, Ian. (2016). *Software Engineering 10<sup>th</sup> Edition*. London: Pearson Education Limited.
- Sulistiyanto, H & Azhari, S.N. (2014). Urgensi Pengujian Pada Kemajemukan Perangkat Lunak dalam Multi Perspektif. *KomuniTi*. 01(01), 65-74 <https://doi.org/10.23917/komuniti.v6i1.2944>
- Uminingsih, Ichsanudin, M. N., Yusuf, M. Suraya. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan dengan Metode Black Box Testing bagi Pemula. *Jurnal Ilmiah Teknik dan Ilmu Komputer (STORAGE)*, 01(02). 1-8
- Wahyu, M. T., Afrizal, M,. (2023). Pengujian Blackbox Metode Equivalent Partitions pada Aplikasi Karyawan Website Oby Komputer. *Jurnal Sistem Informasi (Teknofile)*, 01(02). <https://jurnal.nawansa.com/index.php/teknofile/article/view/16/12>
- Wardhani, D. R., & Abdillah, R. (2018). Pengambilan Keputusan dengan Metode Perbandingan Eksponensial (MPE) dalam Manajemen Kedai. *Semnasristek 2018*, (pp. 439-444). <https://osf.io/preprints/osf/rx2p5>