

# Pengujian Aplikasi Pengelolaan Laundry Menggunakan Metode Blackbox Testing dan Whitebox Testing

Ma'ruf Nur Muhammad<sup>1</sup>, Muhammad Alwan Nurdin<sup>2</sup>, Hanifah Permatasari<sup>\*</sup>

<sup>1,2,3</sup>Sistem Informasi

Universitas Duta Bangsa Surakarta

<sup>1\*</sup>marufnur27@gmail.com, <sup>2</sup>alwannurdinm@gmail.com, <sup>3\*</sup>hanifah\_permatasari@udb.ac.id

*Abstrak*— Pengujian sistem adalah proses penting dalam pengembangan perangkat lunak yang bertujuan untuk memastikan bahwa sistem yang dibangun berjalan sesuai dengan harapan dan memenuhi persyaratan yang ditentukan. Dalam jurnal ini, kami menyajikan tinjauan tentang pengujian sistem, termasuk metode yang biasa digunakan, masalah yang dihadapi, dan keuntungan yang dapat diperoleh dari pengujian sistem yang berhasil. Aplikasi yang tidak layak dan tidak berkualitas dapat mempersingkat waktu pengembangan perangkat lunak serta mengganggu penggunaannya. Proses pengujian bertujuan untuk menemukan kesalahan perangkat lunak, dan setiap aplikasi tidak akan lepas darinya. Blackbox testing fokus utamanya adalah input dan output sistem. Metode ini melihat sistem sebagai "kotak hitam", di mana hanya hasil yang dilihat tanpa memperhatikan detail implementasi. Kami memberikan penjelasan tentang pengujian blackbox yang termasuk pengujian fungsional, kesalahan, keamanan, dan stres. Kelebihan pengujian blackbox termasuk kemampuan untuk menemukan kesalahan fungsional, menguji dari sudut pandang pengguna, dan membedakan pengujian antara pengembang dan penguji. Sedangkan whitebox testing merupakan dimana desain dan pelaksanaan pengujian didasarkan pada pemahaman tentang struktur internal sistem. Metode ini membutuhkan pemahaman tentang aliran program, kode sumber, dan rute yang dapat digunakan saat eksekusi. Kami membahas berbagai metode pengujian whitebox, seperti unit, integrasi, dan jalur.

*Kata kunci*— Pengujian sistem, Whitebox Testing, Blackbox Testing

*Abstract*— System testing is an important process in software development that aims to ensure that the system built runs as expected and meets the specified requirements. In this journal, we provide an overview of system testing, including the methods commonly used, the problems encountered, and the benefits that can be derived from successful system testing. Improper and unqualified applications can shorten software development time and annoy users. The testing process aims to find software errors, and every application is inseparable from them. Blackbox testing focuses primarily on system inputs and outputs. This method sees the system as a "black box", where only results are viewed without regard to implementation details. We provide an explanation of blackbox testing which includes functional, error, security, and stress testing. The advantages of blackbox testing include the ability to find functional errors, test from the user's point of view, and differentiate tests between developers and testers. While whitebox testing is where the design and implementation of testing is based on an understanding of the internal structure of the system. This method requires an understanding of the program flow, source code, and routes that can be used during execution. We cover various whitebox testing methods, such as units, integrations, and paths.

*Keywords*— System testing, Whitebox Testing, Blackbox Testing

## I. PENDAHULUAN

Pengembangan aplikasi perangkat lunak telah menjadi komponen penting dalam industri teknologi informasi di era digital yang semakin maju. Aplikasi perangkat lunak yang sukses dirilis harus memenuhi persyaratan fungsional dan memberikan pengalaman pengguna yang handal dan baik. Untuk mencapai tujuan ini, pengujian aplikasi telah berkembang menjadi bagian penting dari siklus pengembangan perangkat lunak. Analisa, perancangan, implementasi, uji coba, dan pengelolaan adalah beberapa tahapan dalam proses pengembangan sistem atau aplikasi.

Proses uji coba yang buruk dapat meningkatkan biaya pengembangan dan memperpanjang waktu pengembangan aplikasi. Pengujian perangkat lunak memiliki fungsi penting dalam proses pengembangan aplikasi, dengan melakukan pengujian akan didapat seperti galat atau error, untuk mengetahui kualitas perangkat lunak tersebut apakah sudah memenuhi persyaratan kinerja yang

baik dan menentukan perbedaan dari hasil yang diharapkan dengan hasil sebenarnya[1].

Aplikasi yang akan diuji adalah Aplikasi Pengelolaan Laundry, pengujian Aplikasi Pengelolaan Laundry adalah langkah penting dalam siklus pengembangan perangkat lunak. Aplikasi ini harus dirancang dengan baik, mudah digunakan, dan berfungsi dengan baik untuk mendukung kegiatan sehari-hari bisnis laundry. Untuk memastikan bahwa aplikasi ini memenuhi standar kualitas yang diharapkan, pengujian aplikasi pengelolaan laundry melibatkan serangkaian proses pengujian yang bertujuan untuk mengidentifikasi kesalahan atau kelemahan dalam kinerja dan fungsi aplikasi. Ada banyak metode yang digunakan untuk menguji aplikasi secara menyeluruh.

Ada dua jenis software testing yaitu software testing techniques dan software testing strategies, software techniques dibagi menjadi dua teknik pengujian yaitu teknik whitebox testing dan blackbox testing, sedangkan untuk dsoftware

testing strategies diklasifikasikan menjadi tiga yaitu unit testing, integration testing dan validation testing. Dalam jurnal ini akan membahas secara rinci dua metode pengujian yang paling umum digunakan, yaitu blackbox testing, dan whitebox testing.

Pengujian whitebox (pengujian secara sistematis) dan Pengujian blackbox (pengujian secara fungsional) [2]. Tujuan penelitian ini melakukan pengujian sebuah form login menggunakan metode White Box Testing dengan teknik Basis Path. Teknik Basis Path ini digunakan untuk mengetahui kompleksitas logika aplikasi [3].

## II. METODOLOGI PENELITIAN

Berdasarkan pembahasan di atas metode yang digunakan adalah metode blackbox testing dan whitebox testing.

### A. Blackbox Testing

Black box testing merupakan pengujian yang umumnya berkaitan dengan memverifikasi bahwa sistem dapat berfungsi dengan benar dari perspektif pengguna. Pengujian ini biasanya tidak dapat melakukan verifikasi proses sistem internal dan hanya hasil actual yang terlihat oleh pengguna sistem[4].

Pengujian blackbox testing biasanya bertujuan untuk memastikan bahwa sistem dapat beroperasi dengan benar dari sudut pandang pengguna, tetapi biasanya tidak dapat memverifikasi proses sistem internal, hanya menunjukkan hasil nyata yang dapat dilihat pengguna.

### B. Whitebox Testing

White Box Testing yaitu prosedur pengecekan audit yang mengaitkan tilikan mendalam pada bentuk dalam aplikasi. Dalam prosedur ini, pengecekan audit dijalani dengan meninjau serta mengkaji pangkal aplikasi, peredaran pengawasan, dan juga bentuk data yang dipakai[5]. White Box Testing menguatkan kita buat meninjau serta menilai tiap bagian aplikasi sebagai rinci, akibatnya sanggup mengenali kelemahan, cacat fungsional, serta retakan keamanan yang boleh jadi berlangsung[6].

White Box sendiri mempunyai beberapa teknik di dalam pengujiannya, seperti : Data Flow Testing, Control Flow Testing, Basic Path / Path Testing, dan Loop Testing [7]. Dalam Pengujian White Box para penguji perlu mengetahui secara dalam source code yang akan diuji. Pengujian White Box dapat mengungkap kesalahan implementasi dari sebuah aplikasi. Pengujian ini dapat diterapkan pada tingkatan integrasi, unit dan sistem [8].

### C. Teknik Basis Path

Salah satu teknik Pengujian White Box yang pertama kali diusulkan oleh Tom McCabe adalah Teknik Basis Path. Teknik ini memungkinkan penguji untuk mengukur kompleksitas logika dari rancangan prosedural. Skenario uji coba yang dibuat untuk menguji Teknik Basis Path ini dijamin akan mengeksekusi setiap statement dari aplikasi yang diujikan setidaknya satu kali saat tahap pengujian [9].

### D. Cyclomatic Complexity

Cyclomatic Complexity merupakan besaran perangkat lunak yang menyediakan acuan kuantitatif kompleksitas suatu logika dalam program. Rumus menghitung Cyclomatic Complexity adalah sebagai berikut: [10]

$$V(G) = E - N + 2$$

Keterangan :

E = jumlah edges pada flowgraph

N = jumlah nodes pada flowgraph

P = jumlah predicates nodes pada flowgraph

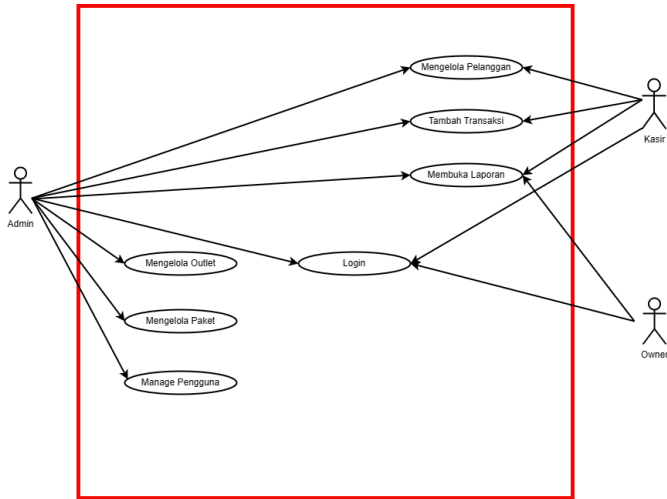
### E. Flowgraph

Flowgraph adalah representasi grafis dari aliran atau urutan operasi dalam program atau sistem. Biasanya digunakan untuk menunjukkan aliran data, aliran kontrol, atau kombinasi keduanya dalam suatu proses atau algoritma. Ini membantu pengembang perangkat lunak atau insinyur sistem memahami dan menganalisis aliran kerja dari awal hingga akhir.

## III. HASIL DAN PEMBAHASAN

### A. Diagram Use Case

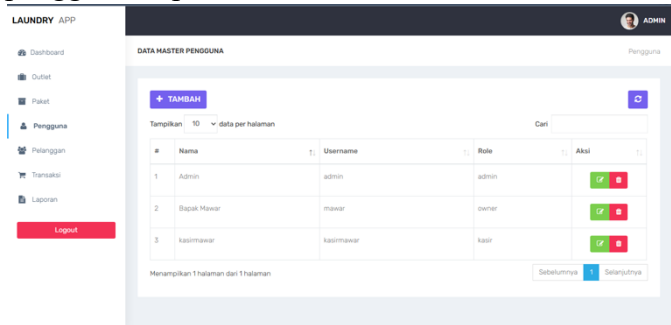
Diagram use case dibawah ini menggambarkan aktor yang terlibat dalam Aplikasi Pengelolaan Laundry yaitu admin, owner, dan kasir. Admin bisa login dan mengakses semua menu, owner hanya bisa login dan memantau laporan, sedangkan kasir dapat mengakses menu pelanggan, transaksi, dan laporan yang dapat dilihat pada Gambar 1.



Gambar 1 Usecase Sistem Laundry.

**B. Blackbox Testing**

Dalam pengujian Blackbox Testing ini halaman atau interface yang akan diuji yaitu halaman pengguna seperti Gambar 2 dibawah ini.



Gambar 2. Interface Pengguna

Form halaman pengguna, from yang dapat berfungsi sebagai menambahkan pelanggan secara manual yang dimana seorang admin menginputkan data pengguna, edit data pengguna, dan hapus data pengguna di Aplikasi Pengelolaan Laundry, tampilan pada Gambar 2.

Tabel 1. Pengujian Halaman Pengguna

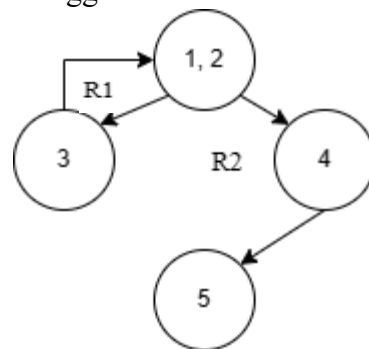
Kode	Rencana Pengujian	Hasil Yang Diharapkan
P01	Menekan tombol	Muncul form penginputan

	“tambah”	data
P02	Menekan tombol tambah mengisi semua input secara benar, lalu menekan tombol simpan	Data akan tersimpan
P03	Menekan tombol tambah, hanya mengisi satu data input, lalu menekan tombol simpan	Muncul peringatan Masukkan semua data
P04	Menekan tombol tambah, salah satu data input tidak sesuai lalu menekan tombol simpan	Muncul pesan Masukkan data yang benar
P05	Menekan tombol “edit”	Muncul form edit data
P06	Menekan tombol edit, mengubah salah satu data dengan benar, lalu menekan tombol simpan	Data berhasil disimpan
P07	Menekan tombol edit, mengubah salah satu data dikosongkan, lalu menekan tombol simpan	Muncul pop up pesan, masukan semua data
P08	Menekan tombol “hapus”	Data akan terhapus

**C. Whitebox Testing**

Pengujian Whitebox Testing dilakukan untuk memeriksa semua jalur dan kondisi yang mungkin telah diuji dengan baik. dan kali ini halaman pengguna yang akan diuji pada tambah pengguna, edit pengguna, dan hapus pengguna.

**a. Tambah Pengguna**



Gambar 3. Flowgraph Tambah Pengguna

Dari gambar 3 merupakan flowgraph dari tambah pengguna, dimana proses tambah data pengguna mempunyai 2 region, 4 (N) node dan 4 (E) edge. Maka untuk Path :  $V(G) = E - N + 2 = 4 - 4 + 2 = 2$ , Maka jumlah Path sebanyak 2.

Untuk jalur independent dari flowgraph edit pengguna, seperti dibawah ini :

Jalur 1 : 1,2,3,1

Jalur 2 : 1,2,4,5

Bukan Jalur Independen : 1,2,3,1,2,4,5

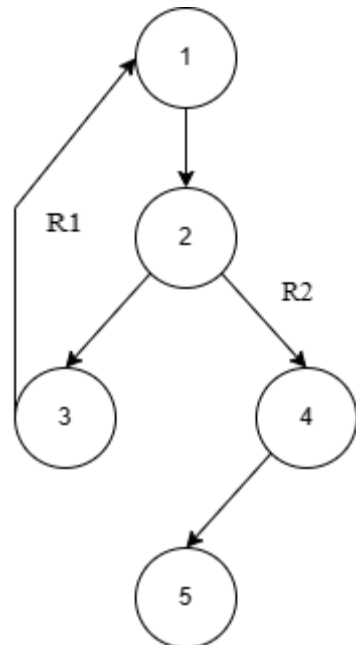
Dibawah ini merupakan matrik grafik dan matrik koneksi dari tambah pengguna.

	1	2	3	4
1		a		
2		c	e	
3				e
4				

Gambar 4. Matrik Grafik Tambah Pengguna

	1	2	3	4	Total
1		a			$1 - 1 = 0$
2		c	e		$2 - 1 = 1$
3				e	$1 - 1 = 0$
4					
Total Kompleksitas Siklomatis					$1 + 1 = 2$

Gambar 8. Matrik Koneksi Tambah Pengguna



Gambar 6. Flowgraph Edit Pengguna

Dari gambar 4 merupakan flowgraph dari edit pengguna, dimana proses edit data pengguna mempunyai jumlah region 2, 5 (N) node dan 5 (E) edge. Maka untuk Path :  $V(G) = E - N + 2 = 5 - 5 + 2 = 2$ , Maka jumlah Path sebanyak 2.

Untuk jalur independen dari flowgraph edit pengguna, seperti dibawah ini :

Jalur 1 : 1,2,3,1

Jalur 2 : 1,2,4,5

Bukan Jalur independen : 1,2,3,1,2,4,5.

Dibawah ini merupakan matrik grafik dan matrik koneksi dari edit pengguna.

	1	2	3	4	5
1		a			
2			b	c	
3	d				
4					e
5					

Gambar 7 Matrik Grafik Edit Pengguna.

## b. Edit Pengguna

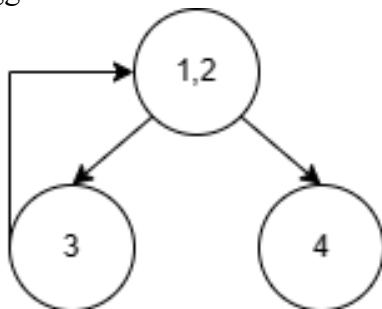
	1	2	3	4	5	Total
1		a				$1 + 1 = 0$
2			b	c		$2 - 1 = 1$
3	d					$1 - 1 = 0$
4					e	$1 - 1 = 0$
5						
<b>Total Kompleksitas Siklomatis</b>						$1 + 1 = 2$

Gambar 8. Matrik Koneksi Edit Pengguna

	1	2	3	Total
1		a	b	$2 - 1 = 1$
2	b			$1 - 1 = 0$
3				
<b>Total Kompleksitas Siklomatis</b>				$1 + 1 = 2$

Gambar 11. Matrik Koneksi Hapus Pengguna

c. Hapus Pengguna



Gambar 9. Flowgraph Hapus Pengguna

Dari gambar 5 merupakan flowgraph hapus pengguna yang memiliki 2 region, 3N(node), 3E(edge). Maka jumlah pathnya 2.

Untuk jalur independen dari Flowgraph hapus pengguna yaitu

Jalur 1 : 1,2,3,1

Jalur 2 : 1,2,4,1

Bukan jalur independent : 1,2,3,1,2,4.

Dibawah ini matrik grafik dan matrik koneksi dari hapus pengguna.

	1	2	3
1		a	b
2	b		
3			

Gambar 10. Matrik Grafik Hapus Pengguna.

IV. KESIMPULAN

Metode pengujian ini memiliki peran penting dalam proses pengujian perangkat lunak. Whitebox testing lebih fokus pada aspek internal perangkat lunak dan membantu mengidentifikasi masalah yang terkait dengan struktur dan logika, sedangkan blackbox testing lebih berfokus pada perilaku eksternal dan pengalaman pengguna. Kombinasi keduanya dapat memberikan cakupan pengujian yang lebih luas dan membantu memastikan perangkat lunak yang handal, berkualitas, dan sesuai dengan persyaratan yang ditetapkan.

Dan juga kedua metode pengujian ini memiliki kelebihan dan kekurangan masing-masing. Whitebox testing lebih mendalam dan dapat mengungkapkan masalah yang terkait dengan implementasi perangkat lunak. Sementara itu, blackbox testing lebih berfokus pada persyaratan fungsional dan perilaku pengguna, sehingga dapat mengidentifikasi masalah yang lebih relevan dengan pengalaman pengguna.

Pilihan metode pengujian yang tepat tergantung pada tujuan pengujian, sumber daya yang tersedia, dan tingkat pengetahuan yang dimiliki oleh tim pengujian. Kombinasi dari kedua metode ini, yang dikenal sebagai greybox testing, juga bisa digunakan untuk menggabungkan keuntungan dari masing-masing metode dan meningkatkan efektivitas pengujian perangkat lunak secara keseluruhan.

REFERENSI

- [1] Pratama, B. P., Ristiano, I. B., Prayogo, I. A., Nasrullah, & Saifudin, A. (2020). Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box
- [2] Y. Irawan, "Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing," *Sentra Penelit. Eng. dan Edukasi*, vol. 9, no. 3, pp. 59–63, 2017.
- [3] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 111, 2020, doi: 10.32493/informatika.v5i2.4713.
- [4] S. Nidhra, "Black Box and White Box Testing Techniques - A literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29-50, 2012
- [5] R. Tamara Aldisa, "Aplikasi Pengolahan Data Penjualan Pembangkit Listrik Tenaga Surya Menggunakan Model View Controler Berbasis Framework CodeIgniter Dan White Box Testing," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, pp. 677–686, 2021.
- [6] J. Publikasi and T. Informatika, "PERANCANGAN APLIKASI PERSEDIAAN BAHAN KUE BERBASIS WEBSITE," vol. 2, no. 2, 2023.
- [7] V. P. Katiyar and S. Patel, "White-Box Testing Technique for Finding Defects," *Glob. J. Res. Anal.*, vol. 8, no. 7, pp. 83–85, 2019, [Online]. Available: <http://worldwidejournals.co.in/index.php/gjra/article/view/4883>.
- [8] K. Mohd. Ehmer and K. Farmeena, "A Comparative Study of White Box , Black Box and Grey Box Testing Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 12–15, 2012.
- [9] Handy and J. Susilo, "Aplikasi Pengujian White-Box Ibi Online Judge," *J. Inform. dan Bisnis*, vol. 3, pp. 56–68, 2014.
- [10] E. sita Eriana, "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Dengan White Box Testing," *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 28–33, 2020.