

**PENGUJIAN BLACK BOX PADA APLIKASI OMNICHANNEL DENGAN
METODE EQUIVALENCE PARTITIONING DAN BOUNDARY VALUE
ANALYSIS**

(Studi Kasus : PT Guardia Teknologi (Jubelio))

(Skripsi)

Oleh

RAHEL ANASTASYA PITHALOKA

2015061086



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

2024

**PENGUJIAN BLACK BOX PADA APLIKASI OMNICHANNEL DENGAN
METODE EQUIVALENCE PARTITIONING DAN BOUNDARY VALUE
ANALYSIS**

(Studi Kasus : PT Guardia Teknologi (Jubelio))

Oleh

RAHEL ANASTASYA PITHALOKA

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai
Gelar Sarjana Teknik**

Pada

Program Studi S1 Teknik Informatika



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

2024

ABSTRAK

PENGUJIAN BLACK BOX PADA APLIKASI OMNICHANNEL DENGAN METODE EQUIVALENCE PARTITIONING DAN BOUNDARY VALUE ANALYSIS

Oleh

RAHEL ANASTASYA PITHALOKA

Omnichannel Jubelio adalah suatu platform yang dapat mengintegrasikan konsumen dengan banyak *channel*. Aplikasi ini terintegrasi ke seluruh channel penjualan *offline* dan *online*, seperti *marketplace*, *social commerce*, *website*, dan toko fisik hanya melalui satu *dashboard*. *Omnichannel* Jubelio memiliki fitur unggulan yaitu *Warehouse Management System* (WMS) yang berfungsi dalam mengelola persediaan, memproses pesanan, dan mengatur alur barang dari masuk hingga keluar gudang. Dengan pertumbuhan pesat pengguna platform Jubelio atau pengembangan fitur-fitur lain, muncul potensi masalah teknis seperti bug yang dapat menghambat pengalaman pengguna dan tantangan dalam menjaga kualitas layanan. Oleh karena itu, pengujian aplikasi sangat penting untuk memastikan skalabilitas sistem dan pemeliharaan kualitas sistem manajemen gudang khususnya pada fitur Proses Pesanan dalam menu Gudang. Penelitian ini menggunakan pengujian *Black Box* dengan kombinasi metode *Equivalence Partitioning* dan *Boundary Value Analysis*. Pengujian black box pada aplikasi omnichannel Jubelio menerapkan metode EP untuk pengujian yang lebih efisien karena hanya satu atau beberapa nilai dari setiap partisi yang diuji. Sedangkan metode BVA untuk memastikan bahwa sistem dapat menangani nilai-nilai ekstrem di batas-batas input dengan baik. Berdasarkan hasil pengujian aplikasi yang dilakukan pada delapan fitur aplikasi dengan total 95 skenario pengujian, terdapat 82 skenario yang berhasil, dan 13 skenario yang gagal. Nilai efektivitas aplikasi yang didapat dari pengujian tersebut sebesar 86,31% yang artinya memiliki nilai efektivitas yang cukup. Selain itu, terdapat lima rekomendasi perbaikan yang dapat diberikan kepada tim pengembang.

Kata kunci : *Black Box*, *Boundary Value Analysis*, *Equivalence Partitioning*, *Omnichannel*, *Warehouse Management System*

ABSTRACT

BLACK BOX TESTING ON OMNICHANNEL APPLICATIONS WITH EQUIVALENCE PARTITIONING AND BOUNDARY VALUE ANALYSIS METHODS

By

RAHEL ANASTASYA PITHALOKA

Omnichannel Jubelio is a platform integrating consumers with multiple channels. This application is integrated into all offline and online sales channels, such as marketplaces, social commerce, websites, and physical stores through one dashboard. Omnichannel Jubelio has a superior feature, namely the Warehouse Management System (WMS) which functions to manage inventory, process orders, and regulate the chain process of goods from entering to leaving the warehouse. With the rapid growth of Jubelio platform users or the development of other features, there is the potential for technical problems such as bugs that can hinder user experience and challenges in maintaining service quality. Therefore, application testing is very important to ensure system scalability and maintain the quality of the warehouse management system (WMS), especially in the Order Process feature in the Warehouse menu. In research Black Box testing is used combination of the Equivalence Partitioning and Boundary Value Analysis methods. Black box testing on the Omnichannel Jubelio applies the EP method for more efficient testing, as only one or a few values from each partition are tested. The BVA method is used to ensure that the system can handle extreme values at input boundaries effectively. The result show that application testing carried out on eight application features with a total of 95 test scenarios, there were 82 successful scenarios and 13 failed scenarios. The application effectiveness value obtained from the test was 86.31%, which means it has a sufficient effectiveness value, futhermore, five recommendations for improvement that can be given to the development team.

Keywords: Black Box, Boundary Value Analysis, Equivalence Partitioning, Omnichannel, Warehouse Management System

Judul Skripsi : **PENGUJIAN BLACK BOX PADA APLIKASI
OMNICHANNEL DENGAN METODE
EQUIVALENCE PARTITIONING DAN
BOUNDARY VALUE ANALYSIS**

Nama Mahasiswa : **Rahel Anastasya Pithaloka**

Nomor Pokok Mahasiswa : **2015061086**

Program Studi : **Teknik Informatika**

Fakultas : **Teknik**

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.
NIP. 197216031999032002


Ir. Titin Yulianti, S.T., M.Eng.
NIP. 198807092019032015

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

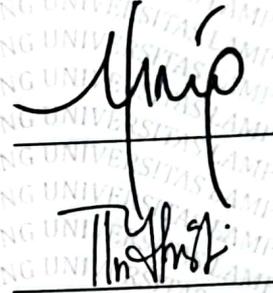

Herlinawati, S.T., M.T.
NIP. 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Eng. Ir. Mardiana, S.T.,M.T., IPM.**



Sekretaris : **Ir. Titin Yulianti, S.T., M.Eng.**



Penguji : **Ir. M. Komarudin, S.T.,M.T.**



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.)
NIP. 197509282001121002



Tanggal Lulus Ujian Skripsi: 14 Agustus 2024

PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Pengujian Black Box pada Aplikasi Omnichannel dengan Metode Equivalence Partitioning dan Boundary Value Analysis” adalah hasil karya saya sendiri. Seluruh hasil skripsi ini telah mengikuti pedoman penulisan karya ilmiah Universitas Lampung. Jika di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dikerjakan oleh orang lain, maka saya siap menerima sanksi sesuai dengan ketentuan hukum dan akademik yang berlaku.

Bandar Lampung, 20 Agustus 2024

Pembuat Pernyataan,



Rahel Anastasya Pihaloka

NPM. 2015061086

RIWAYAT HIDUP



Penulis dilahirkan di Jakarta pada tanggal 28 Juni 2002, sebagai anak ketiga dari Ayah bernama Terry Ramses Simanjuntak dan Ibu Artia Minar Simatupang. Penulis menyelesaikan pendidikan formal dimulai dari TK Anugerah Kudus pada tahun 2008. Selanjutnya Pendidikan dasar di SD Tarsisius Vireta yang selesai pada tahun 2014. Pada tahun 2017 menyelesaikan pendidikan menengah pertama di SMP Negeri 5 Pasar Kemis, kemudian melanjutkan pendidikan menengah atas di SMA Negeri 96 Jakarta pada tahun 2020.

Pada tahun 2020 melanjutkan pendidikan sebagai mahasiswi di Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswi, penulis melakukan beberapa kegiatan, yaitu sebagai berikut :

1. Menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) Universitas Lampung, Departemen Pendidikan dan Pengembangan Diri, Divisi Kerohanian periode 2021,
2. Menjadi anggota Panitia Publikasi Dekorasi dan Dokumentasi (PPD) dalam acara Electrical Engineering in Action oleh HIMATRO Universitas Lampung pada tahun 2021,
3. Menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) Universitas Lampung, Departemen Pendidikan dan Pengembangan Diri, Divisi Minat dan Bakat periode 2022,
4. Mengikuti program Studi Independen Bersertifikat Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Full Stack Web Development di PT Arkatama Multi Solusindo pada tahun 2022,

5. Mengikuti kegiatan Magang Mandiri sebagai Quality Assurance Intern di PT Guardia Teknologi Jubelio pada tahun 2022,
6. Mengikuti program Studi Independen Bersertifikat Kampus Merdeka dari Kementrian Pendidikan dan Budaya dengan mengambil Web Frontend Engineer di Productzilla Academy pada tahun 2023,
7. Menjadi karyawan full time di PT Guardia Teknologi Jubelio sebagai Junior Quality Assurance hingga saat ini,
8. Mengikuti kegiatan Kuliah Kerja Nyata di Desa Sukadamai, Kec. Gunung Alip, Kabupaten Tanggamus, Lampung pada bulan Januari 2024 sampai Febuari 2024.

PERSEMBAHAN

Segala puji dan syukur kepada Tuhan Yesus Kristus yang memberikan kekuatan, berkat, dan pertolongannya melalui orang-orang yang membimbing dan mendukung penulis dalam penyusunan skripsi ini.

Kupersembahkan karyaku ini untuk orang-orang yang kusayangi dengan setulus hati:

Mama dan Papa,

Yang selalu memberikan cinta, kasih sayang, dan doa yang tiada henti kepada anaknya, serta selalu menguatkan penulis dalam masa sulit penulis.

Keluargaku,

Orang yang penulis sayangi yang selalu memberi semangat dan dukungan dalam bentuk apapun serta doa sampai saat ini.

Dan

Diri sendiri,

Yang telah berjuang dan berusaha sampai akhir. Terima kasih karena telah kuat dan tidak menyerah atas segala rintangan yang terjadi. Terima kasih atas setiap keringat dan tetes mata yang berjatuhan selama perjuangan ini. Terima kasih telah membuktikan bahwa kita mampu mencapai titik ini. Mari tetap berdoa dan berjuang untuk masa depan yang cerah kelak.

Terimakasih untuk segalanya.

MOTTO

In The Name of Jesus Christ

“Direndahkan di mata manusia, Ditinggikan di mata Tuhan.”

Aku ditolak dengan hebat sampai jatuh, tetapi TUHAN menolong aku.

(Mazmur 118 : 13)

Sekalipun aku berjalan dalam lembah kekelaman, aku tidak takut bahaya, sebab Engkau besertaku; gada-Mu dan tongkat-Mu, itulah yang menghibur aku.

(Mazmur 23 : 4)

Aku tahu, bahwa Engkau sanggup melakukan segala sesuatu, dan tidak ada rencana-Mu yang gagal.

(Ayub 42 : 2)

Apa pun juga yang kamu perbuat, perbuatlah dengan segenap hatimu seperti untuk Tuhan dan bukan untuk manusia.

(Kolose 3 : 23)

Karena masa depan sungguh ada, dan harapanmu tidak akan hilang.

(Amsal 23 : 18)

SANWACANA

Puji syukur dan hormat kepada Tuhan Yesus Kristus yang telah menyertai dan memberi berkat sepanjang hidup penulis, sehingga penulis dapat menyelesaikan proses penyusunan skripsi yang berjudul “Pengujian Black Box pada Aplikasi Omnichannel dengan Metode Equivalence Partitioning dan Boundary Value Analysis”. Selama proses penyusunan skripsi ini, penulis telah mendapatkan berbagai pengalaman berharga, bantuan yang berharga, dan panduan yang sangat membantu dari berbagai pihak. Maka dari itu, penulis ingin mengucapkan terima kasih kepada semua pihak yang membantu, khususnya kepada :

1. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Program Jurusan Teknik Elektro Universitas Lampung.
3. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu dalam proses pengerjaan skripsi.
4. Ibu. Dr. Eng. Ir. Mardiana, S.T., M.T., IPM., selaku Pembimbing Utama yang telah memberi waktu, tenaga, dan pikiran untuk membimbing dan mendukung penulis dalam penyusunan skripsi ini.

5. Ibu Ir. Titin Yulianti, S.T., M.Eng., selaku Pembimbing Pendamping yang selalu meluangkan waktu dan memberi tenaga untuk membimbing dan mendukung penulis dalam penyusunan skripsi ini.
6. Bapak Ir. M. Komarudin, S.T.,M.T., selaku Dosen Penguji yang telah membimbing saya dalam penyusunan skripsi ini.
7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Pembimbing Akademik yang selalu meluangkan waktunya untuk membimbing dan mendukung selama perkuliahan.
8. Mama dan Papa tercinta, Ibu Artia Minar Simatupang dan Bapak Terry Ramses Simanjuntak yang walaupun tidak sempat mencapai gelar perkuliahan namun beliau mampu mendidik penulis agar dapat melewati setiap hambatan dan rintangan sehingga penulis boleh berada di titik akhir ini. Khususnya, terima kasih karena telah menguatkan penulis saat penulis berada di titik terendah dan mau mengantar dan menemani penulis pulang pergi Tangerang – Lampung selama proses penyusunan skripsi ini. *Thank you for everything* Pa, Ma! Gelar ini kupersembahkan untuk kalian.
9. Abang dan Adik yang sangat kusayangi, Paskah, Mikha, dan Elizabeth. Terima kasih atas dukungan semangat dan doa nya, serta bantuan-bantuan apapun yang menyertai penulis sampai saat ini,
10. Mbak Rika selaku Admin Program Studi Teknik Informatika yang banyak membantu penulis dalam segala urusan administrasi selama perkuliahan.
11. PT Guardia Teknologi (Jubelio) yang telah menjadi tempat dan memberi peluang kepada saya untuk menjalani penelitian.

12. KKN Sukadamai *Gengs*, yang telah mendukung dan menemani saya serta mendengarkan keluh kesah saya selama proses penyusunan skripsi.
13. Teman-teman tersayang, baik teman-teman perkuliahan yang membantu membimbing penulisan skripsi, teman-teman rumah dan gereja yang selalu memberi dukungan dan semangat.
14. Terakhir, kepada diri sendiri, Rahel Anastasya Pithaloka. Terima kasih sudah bertahan sejauh ini. Terima kasih sudah berusaha dan merayakan diri sendiri sampai di titik ini. Walaupun seringkali merasa putus asa dan ingin menyerah, namun terima kasih karena telah menjadi manusia yang selalu mau berusaha. Terima kasih karena memutuskan untuk tidak menyerah sesulit apapun dalam proses penyusunan skripsi sambil bekerja dengan jarak Tangerang – Lampung, berat, tapi kamu berhasil menyelesaikannya. Lagi dan lagi, bukan karena kuat dan hebatku, tetapi karena Tuhan yang bekerja. Tetap semangat dalam masa depan cerah yang menanti!

Sekali lagi, penulis ucapkan terima kasih kepada semua dan pihak yang tidak bisa disebutkan satu persatu. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi penulis ataupun pihak lain yang membaca.

Bandar Lampung,
Penulis,

Rahel Anastasya Pithaloka
NPM. 2015061086

DAFTAR ISI

	Halaman
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	4
II. TINJAUAN PUSTAKA.....	6
2.1 <i>Omnichannel</i> Jubelio	6
2.2 Pengujian (<i>Testing</i>)	9
2.3 <i>Software Quality Assurance</i>	10
2.4 <i>Blackbox Testing</i>	14
2.5 <i>Test Case</i>	16
2.6 <i>Equivalence Partitioning</i>	17
2.7 <i>Boundary Value Analysis</i>	20
2.8 <i>Software Testing Life Cycle (STLC)</i>	23
2.9 Penelitian Terkait	24
III. METODOLOGI PENELITIAN.....	27
3.1 Waktu dan Tempat	27
3.2 Alat dan Bahan	27
3.3 Tahapan Penelitian.....	28
IV. HASIL DAN PEMBAHASAN.....	51
4.1 Cara Kerja Aplikasi	51
4.2 Hasil Pengujian.....	61

4.3	Nilai Efektivitas Aplikasi	81
4.4	Daftar Temuan Aplikasi.....	84
	87
V.	KESIMPULAN DAN SARAN.....	87
5.1	Kesimpulan.....	87
5.2	Saran.....	88
	DAFTAR PUSTAKA	89
	LAMPIRAN A. HASIL UJI FITUR 1	93
	LAMPIRAN B. HASIL UJI FITUR 2	100
	LAMPIRAN C. HASIL UJI FITUR 3	104
	LAMPIRAN D. HASIL UJI FITUR 4	110
	LAMPIRAN E. HASIL UJI FITUR 5	120
	LAMPIRAN F. HASIL UJI FITUR 6	134
	LAMPIRAN G. HASIL UJI FITUR 7	141
	LAMPIRAN H. HASIL UJI FITUR 8	148

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2. 1 Konsep Omnichannel	6
Gambar 2. 2 Tampilan Website Menu Gudang	8
Gambar 2. 3 Tampilan Website Proses Pesanan – Gudang	9
Gambar 2. 4 Diagram Alir Proses SQA	11
Gambar 2. 5 Konsep Black Box Testing	14
Gambar 2. 6 Ilustrasi Metode Equivalence Partitioning	17
Gambar 2. 7 Tahapan Software Testing Life Cycle (STLC).....	23
Gambar 3. 1 Tahapan Penelitian	29
Gambar 4. 1 Diagram Alur Fitur Login User	51
Gambar 4. 2 Halaman Login.....	52
Gambar 4. 3 Halaman Utama.....	52
Gambar 4. 4 Halaman Picking - Belum Mulai.....	53
Gambar 4. 5 Diagram Alur Fitur Serahkan ke Picker	53
Gambar 4. 6 Pop Up Serahkan ke Picker.....	54
Gambar 4. 7 Diagram Alur Fitur Cetak Picklist.....	54
Gambar 4. 8 Pop Up Cetak Picklist	55
Gambar 4. 9 Halaman Packing - Belum Mulai	55
Gambar 4. 10 Diagram Alur Fitur Mulai Packing	56
Gambar 4. 11 Halaman Mulai Packing	56
Gambar 4. 12 Halaman Shipping - Siap Kirim	57
Gambar 4. 13 Diagram Alur Fitur Buat Pengiriman Baru	58
Gambar 4. 14 Halaman Buat Pengiriman Baru.....	58
Gambar 4. 15 Diagram Alur Fitur Buat Pengiriman Terpilih	59
Gambar 4. 16 Halaman Buat Pengiriman Terpilih.....	59
Gambar 4. 17 Diagram Alur Fitur Tambah Pengiriman Baru	60
Gambar 4. 18 Halaman Masukkan Pengiriman	60
Gambar 4. 19 Diagram Alur Fitur Tambah Pengiriman Baru	61

Gambar 4. 20 Halaman Tambah Pengiriman Baru	61
Gambar 4. 21 Grafik Jumlah Skenario Valid dan Invalid EP	82
Gambar 4. 22 Grafik Jumlah Skenario dengan Berbagai Nilai BVA	82
Gambar 4. 23 Grafik Jumlah Skenario Berhasil dan Gagal.....	83

DAFTAR TABEL

Gambar	Halaman
Tabel 2. 1 Batasan Data Boundary Value Analysis	21
Tabel 2. 2 Penelitian Terkait.....	25
Tabel 3. 1 Jadwal Penelitian.....	27
Tabel 3. 2 Skenario Pengujian - Login User	31
Tabel 3. 3 Skenario Pengujian - Cetak Picklist	33
Tabel 3. 4 Skenario Pengujian - Serahkan ke Picker	34
Tabel 3. 5 Skenario Pengujian - Mulai Packing.....	36
Tabel 3. 6 Skenario Pengujian - Buat Pengiriman Baru.....	38
Tabel 3. 7 Skenario Pengujian - Buat Pengiriman Terpilih.....	41
Tabel 3. 8 Skenario Pengujian – Tambah Kartu Pengiriman Baru.....	43
Tabel 3. 9 Skenario Pengujian - Masukkan Pengiriman	45
Tabel 4. 1 Hasil Pengujian - Login User	62
Tabel 4. 2 Hasil Pengujian - Cetak Picklist.....	63
Tabel 4. 3 Hasil Pengujian - Serahkan ke Picker	65
Tabel 4. 4 Hasil Pengujian - Mulai Packing.....	67
Tabel 4. 5 Hasil Pengujian - Buat Pengiriman Baru	70
Tabel 4. 6 Hasil Pengujian - Buat Pengiriman Terpilih.....	74
Tabel 4. 7 Hasil Pengujian - Tambah Pengiriman Baru	75
Tabel 4. 8 Hasil Pengujian - Masukkan Pengiriman	77
Tabel 4. 9 Daftar Temuan Kesalahan Aplikasi.....	85

I. PENDAHULUAN

1.1 Latar Belakang

Dalam era transformasi digital yang sedang berlangsung di Indonesia, sektor *e-commerce* tumbuh pesat dan memberikan potensi besar bagi perusahaan-perusahaan teknologi untuk mengembangkan solusi yang revolusioner. Menanggapi peluang ini, Jubelio, hadir sebagai solusi inovatif dengan menyediakan platform *omnichannel* untuk mendukung pelaku bisnis dalam mengelola operasional mereka di berbagai platform *e-commerce* yang tersebar dengan memungkinkan semua *channel* tersedia untuk konsumen dan saling terhubung atau ter-integrasi satu sama lain. Jubelio dengan url link www.v2.jubelio.com menyediakan fitur unggulan, yaitu *Warehouse Management System* (WMS). Fitur ini menjawab tantangan kompleksitas dalam mengelola persediaan, memproses pesanan, dan mengatur alur barang dari masuk hingga keluar gudang.

Namun, seiring dengan pertumbuhan pesat dalam jumlah pengguna dan pelanggan yang menggunakan platform Jubelio, ada potensi untuk timbulnya berbagai masalah teknis, termasuk ketidaksempurnaan dalam fitur-fitur yang ada atau kemungkinan adanya *bug* yang dapat menghambat pengalaman pengguna. Selain itu, kompleksitas integrasi ini seringkali menimbulkan tantangan dalam menjaga konsistensi dan kualitas layanan. Adapun beberapa aspek yang perlu diperhatikan meliputi skalabilitas sistem, integrasi yang lebih baik dengan *marketplace* yang baru muncul, dan pemeliharaan kualitas sistem manajemen gudang (WMS). Maka dari itu, pengujian pada suatu aplikasi sangat penting untuk dilakukan. Pengujian adalah proses menemukan kesalahan atau cacat pada perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan

kebutuhan pengguna. Sebuah perangkat lunak dianggap tidak memenuhi standar jika tidak sesuai dengan spesifikasi yang telah ditentukan [1]. Dalam proses pengujian perangkat lunak, setiap kasus yang akan diuji harus memiliki identitas dan keterkaitan antara sekumpulan *input-an* dengan *output* yang diinginkan[2].

Penelitian ini merupakan perluasan dari proyek magang peneliti sebelumnya pada PT Guardia Teknologi (Jubelio). Peneliti bertanggung jawab sebagai *Quality Assurance* untuk mengatasi berbagai masalah yang mungkin muncul dalam pengembangan dan pemeliharaan aplikasi *Omnichannel* Jubelio dengan melakukan pengujian aplikasi. Untuk mengatasi masalah tersebut, penelitian ini akan berfokus dalam mengimplementasikan metode pengujian *black box* yaitu pengujian yang dilakukan mengabaikan kode program di dalam sistem tersebut [3]. Pengujian *black box* hanya perlu melihat nilai *output* berdasarkan nilai *input* itu sendiri dengan menerapkan teknik *Equivalence Partitioning* dan *Boundary Value Analysis* pada fitur proses pesanan dalam modul Gudang dari aplikasi *Omnichannel* Jubelio, yang merupakan lingkup WMS (*Warehouse Management System*).

Equivalence Partitioning adalah metode pengujian berdasarkan *input-an* suatu *form* pada aplikasi dengan membagi nilai *input* menjadi valid dan tidak valid dengan tipe data *range* dan bebas. Sedangkan, *Boundary Value Analysis* adalah metode pengujian berdasarkan batas atas dan batas bawah dari nilai *input* suatu *form*, yang berarti hanya dapat menguji tipe data *range*[4]. Dalam penelitian ini akan dilakukan pengujian dengan kedua metode yaitu EP untuk menguji nilai *input* dengan kriteria tertentu, sedangkan BVA untuk menguji nilai *input* dengan rentang tertentu, dan nantinya akan dilakukan perbandingan dari kedua metode tersebut.

Penelitian ini bertujuan untuk mengidentifikasi dan mengatasi cacat dalam aplikasi yang dapat mempengaruhi kualitas layanan *omnichannel*, dengan harapan dapat meningkatkan kepuasan pelanggan dan efisiensi operasional Jubelio. Dengan batasan masalah yang telah ditetapkan, penelitian ini juga diharapkan dapat memberikan wawasan yang berharga tentang bagaimana metode pengujian *Equivalence Partitioning* dan *Boundary Value Analysis* dapat diterapkan untuk meningkatkan kualitas perangkat lunak, khususnya aplikasi *omnichannel*. Dalam

menghitung nilai efektivitas aplikasi, bobot nilai untuk metode EP dan BoBVA biasanya dianggap sama jika tujuan dari pengujian adalah untuk mengukur seberapa baik aplikasi berfungsi secara keseluruhan tanpa memperhitungkan kompleksitas atau detail dari setiap metode pengujian [5]. Namun, jika pengujian ingin memberikan bobot lebih pada metode tertentu berdasarkan kompleksitas atau pentingnya metode tersebut dalam konteks aplikasi yang diuji, maka bobot bisa saja berbeda.

Secara umum, dalam banyak literatur dan praktik pengujian perangkat lunak, kedua metode ini sering dianggap setara bobot nilainya dalam konteks pengujian black box [6]. Tujuan utama dari keduanya adalah untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi tanpa memperhatikan detail implementasi internal. Penelitian ini tidak hanya relevan bagi Jubelio tetapi juga dapat memberikan kontribusi pada literatur pengujian perangkat lunak dan *best practice* dalam industri yang serupa.

1.2 Perumusan Masalah

Adapun permasalahan yang dapat dirumuskan berdasarkan latar belakang yang diuraikan adalah sebagai berikut :

1. Bagaimana implementasi pengujian *black box* dengan metode *equivalence partitioning* dan *boundary value analysis* dalam meningkatkan kualitas aplikasi *Omnichannel* Jubelio pada fitur proses pesanan di menu Gudang?
2. Sejauh mana nilai efektivitas yang didapat dari pengujian metode *equivalence partitioning* dan *boundary value analysis* dalam aplikasi *Omnichannel* Jubelio?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Mengimplementasikan pengujian *black box* dengan metode *equivalence partitioning* dan *boundary value analysis* pada *Omnichannel* Jubelio

2. Menilai efektivitas aplikasi omnichannel Jubelio dengan pengujian *black box* dengan metode *equivalence partitioning* dan *boundary value analysis*.
3. Memberi rekomendasi perbaikan kepada tim pengembang berdasarkan hasil temuan pengujian aplikasi.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Meningkatkan kualitas aplikasi *Omnichannel* Jubelio pada fitur proses pesanan di menu Gudang melalui identifikasi dan perbaikan *bug* serta kesalahan fungsi, alur bisnis, dan antarmuka pengguna yang lebih efisien.
2. Menyediakan dokumentasi butir uji yang dapat digunakan untuk pengujian di masa depan atau sebagai panduan untuk pengembangan fitur baru.
3. Memberikan wawasan tentang efektivitas metode *equivalence partitioning* dan *boundary value analysis* dalam pengujian aplikasi, sehingga dapat dijadikan sebagai bagian dari strategi pengujian perangkat lunak di masa mendatang.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Pengujian dilakukan pada fitur-fitur umum proses pesanan dalam menu Gudang – Proses Pesanan
2. Pengujian dilakukan secara manual tanpa mengetahui kode sumber dan arsitektur sistem di dalamnya
3. Pengujian yang dilakukan terbatas pada pengujian fungsionalitas.

1.6 Sistematika Penulisan

Adapun sistematika penulisan ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan membahas latar belakang, perumusan masalah, tujuan dan manfaat dari penelitian, batasan masalah dalam penelitian, serta sistematika laporan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini membahas teori dasar yang berkaitan dengan proses pengujian ini seperti *Omnichannel*, Pengujian (*Testing*), *Software Quality Assurance*, *Black box Testing*, *Test Case*, *Equivalence Partitioning*, *Boundary Value Analysis*, *Software Testing Life Cycle* (STLC), dan beberapa penelitian terkait.

BAB III : METODOLOGI PENELITIAN

Pada bab ini berisi waktu dan tempat penelitian, alat dan bahan, dan metode yang digunakan dalam penelitian yaitu

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang proses dan hasil pengujian *black box* pada aplikasi *omnichannel* berupa tampilan aplikasi dan skenario pengujian.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan berdasarkan hasil penelitian pengujian aplikasi serta saran yang bermanfaat untuk kualitas yang lebih baik lagi.

II. TINJAUAN PUSTAKA

2.1 *Omnichannel* Jubelio

Omni-channel dapat didefinisikan sebagai manajemen sinergis dari berbagai saluran yang tersedia dan titik kontak pelanggan, sedemikian rupa sehingga pengalaman pelanggan dan kinerja atas saluran yang ada dapat dioptimalkan [7].



Gambar 2. 1 Konsep *Omnichannel*

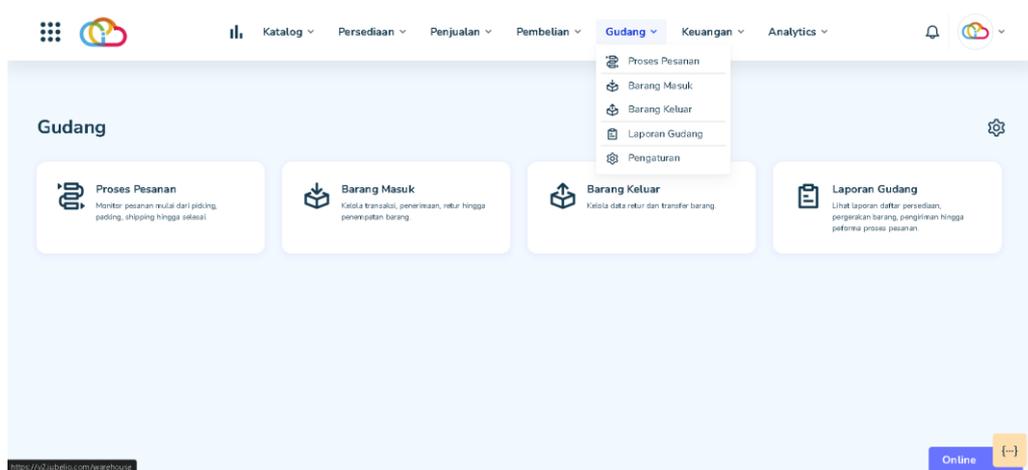
Berdasarkan gambar 2.1, konsep *omnichannel* memungkinkan semua *channel* tersedia untuk konsumen dan saling terhubung atau ter-integrasi satu sama lain. Secara sederhana *omnichannel* adalah pendekatan penjualan multi-saluran yang

berfokus pada memberikan pengalaman pelanggan tanpa batas, baik klien berbelanja *online* dari perangkat seluler, laptop, atau di toko fisik [8]. Penjual dapat berinteraksi dengan pelanggan melalui saluran yang tak terhitung jumlahnya. [9]

Jubelio adalah platform *omnichannel* yang telah dilengkapi banyak fitur untuk memudahkan kelola operasional bisnis *online* dan *offline*. Aplikasi ini sudah terintegrasi ke seluruh channel penjualan *offline* dan *online*, seperti *marketplace*, *social commerce*, *website*, dan toko fisik hanya melalui satu *dashboard*. Sebut saja *marketplace* seperti Shopee, Tokopedia, dkk. Lalu TikTok Shop, Instagram *Shopping* dan Facebook *Marketplace*. Kemudian ada juga toko *online* atau *webstore* seperti, WooCommerce, Shopify dan Jubelio Store. Serta toko *offline* atau point of sales (POS) seperti DealPOS dan Jubelio POS. Jubelio *Omnichannel* dapat melakukan berbagai hal seperti mengurus penjualan harian, *update* stok, menambah produk, menerima dan menempatkan barang yang datang hingga hitung-menghitung kondisi keuangan, semua bisa dilakukan dari Jubelio *Omnichannel*. Dengan begitu pengelolaan dan operasional harian jauh lebih mudah dan praktis, tanpa mengakses satu per satu kanal.

Pada *Omnichannel* Jubelio, terdapat fitur WMS (*Warehouse Management System*) yang merupakan sistem manajemen gudang yang digunakan untuk mengelola seluruh operasional bisnis dalam sebuah gudang secara lengkap. Dalam fitur ini, user dapat mengatur aktivitas penjualan, pembelian, memproses pesanan hingga manajemen persediaan. Penggunaan fitur gudang di Jubelio terdapat 5 bagian, yaitu pengaturan *layout* dan staf gudang, alur singkat penerimaan dan penempatan barang masuk, alur memproses pesanan mulai dari pengambilan, pengepakan, dan pengiriman, serta memperbaiki kesalahan dalam pemrosesan pesanan, mengelola barang keluar mulai dari retur pembelian dan transfer stok antar lokasi, dan cara penyesuaian persediaan dan stok opname. Pada menu Gudang, terdapat sub-menu, yaitu Barang Masuk, Pemrosesan Pesanan, Barang Keluar, serta Penyesuaian Stok

dan Opname. Gambar 2.2 di bawah ini menunjukkan tampilan *website* secara umum menu Gudang pada *Omnichannel Jubelio*.



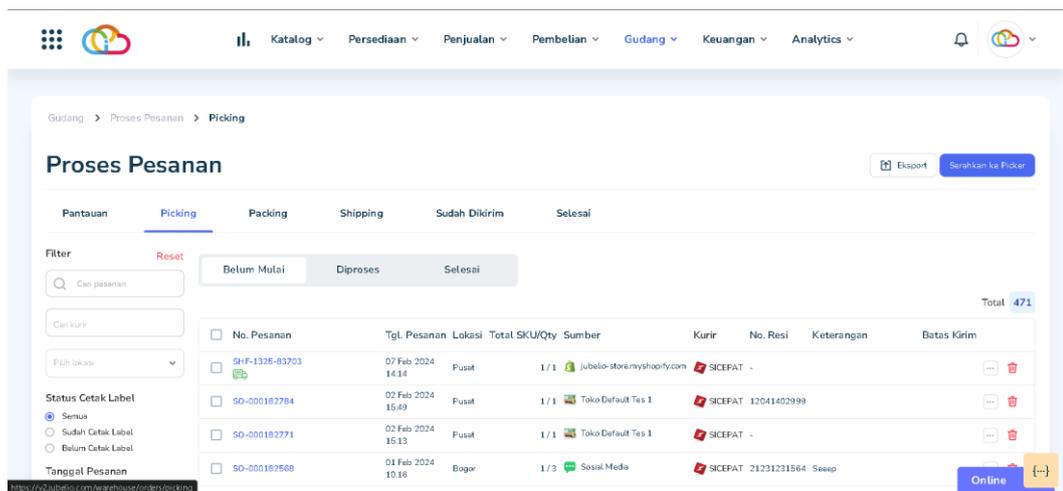
Gambar 2. 2 Tampilan Website Menu Gudang

Penelitian ini berlangsung sebagai keterlanjutan dari proyek magang yang peneliti jalani selama 3 bulan di PT Guardia Teknologi (Jubelio). Dalam periode tersebut, peneliti menjadi seorang *Quality Assurance* fitur WMS / Gudang yang memiliki tanggung jawab sebagai berikut :

- Mengidentifikasi skenario pengujian untuk fitur WMS Jubelio.
- Melakukan pengujian manual terhadap fitur-fitur yang relevan.
- Mendokumentasikan temuan pengujian dengan jelas dan rinci.
- Berkolaborasi dengan tim pengembang untuk memahami perbaikan yang diperlukan.
- Belajar tentang standar kualitas dan praktik pengujian terbaik dalam industri.

Selama periode magang berlangsung, dilakukan pengujian terhadap fitur-fitur umum atau fitur baru di lingkup WMS, serta fokus melakukan pengujian berdasarkan tiket temuan *bug* yang diterima dari *Customer Support* Jubelio. Akan tetapi, pada penelitian ini akan berfokus pada pengujian masukan dan keluaran pada form yang terdapat pada fitur pesanan dalam modul Penjualan dan modul Gudang

yang merupakan lingkup WMS, menggunakan konsep metode *equivalence partitioning* dan *boundary value analysis* untuk memastikan efektivitas dan efisiensi pengujian lebih lanjut. Penelitian dilakukan pada fitur pesanan dalam modul Penjualan dan Gudang dikarenakan fitur tersebut saling berkaitan satu sama lain, dan merupakan fitur penting yang sangat banyak dan sering dipakai oleh klien Jubelio. Berikut adalah tampilan website Jubelio secara umum yang akan diuji :



Gambar 2. 3 Tampilan Website Proses Pesanan – Gudang

Gambar 2.3 merupakan tampilan *website* menu Proses Pesanan pada modul Gudang, yang terdapat beberapa tab seperti : *Picking* (seperti di gambar), *Packing*, *Shipping*, Sudah Dikirim, dan Selesai. Pengujian akan dilakukan pada semua form editor pesanan, form *picking*, form *packing*, serta semua form buat pengiriman pada tab *shipping*.

2.2 Pengujian (*Testing*)

Pengujian (*testing*) merupakan tahapan kritis dalam siklus pengembangan perangkat lunak yang bertujuan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan pengguna. Pengujian adalah proses yang dilakukan untuk menemukan kesalahan atau cacat pada perangkat lunak. Pengujian perangkat lunak sangat penting dalam pengembangan

perangkat lunak karena dapat membantu mengurangi kesalahan dan cacat pada perangkat lunak. Pengujian perangkat lunak dapat dilakukan dengan berbagai cara seperti pengujian fungsional, pengujian non-fungsional, pengujian manual, dan pengujian otomatis.

Menurut IEEE Standard Glossary of Software Engineering Terminology, terdapat dua definisi untuk pengujian yaitu [10]:

1. Proses mengoperasikan sistem atau komponen di bawah kondisi tertentu, mengamati atau mencatat hasilnya, dan melakukan evaluasi terhadap beberapa aspek dari sistem atau komponen.
2. Proses menganalisis item perangkat lunak untuk mendeteksi perbedaan antara kondisi yang ada dan yang dibutuhkan (yaitu, *bug*) dan mengevaluasi fitur item perangkat lunak.

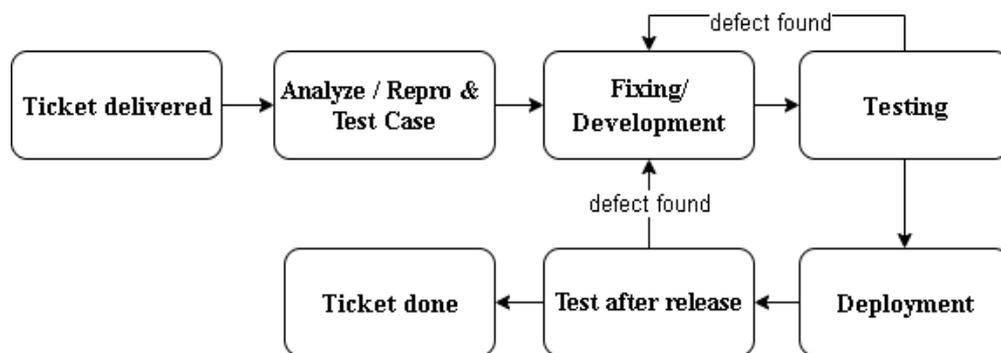
Dalam pengujian perangkat lunak terdapat beberapa jenis pengujian seperti pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional adalah jenis pengujian yang dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan. Sedangkan pengujian non-fungsional adalah jenis pengujian yang dilakukan untuk memastikan bahwa perangkat lunak memenuhi persyaratan non-fungsional seperti keamanan, kinerja, dan keandalan.

2.3 *Software Quality Assurance*

Software Quality Assurance (SQA) merupakan pola terencana dan sistematis yang diperlukan untuk memastikan produk sesuai dengan persyaratan teknis yang telah ditetapkan. SQA juga merupakan rangkaian kegiatan evaluasi yang dirancang untuk mengembangkan atau menghasilkan suatu produk[10].

Quality Assurance Tester atau *Quality Assurance Engineering* adalah para petugas yang bertanggung jawab untuk memastikan kualitas dari *software* atau aplikasi yang diciptakan oleh perusahaan. QA melibatkan pengujian produk atau jasa secara berkala untuk memastikan bahwa produk atau jasa tersebut memenuhi standar

kualitas yang telah ditetapkan. QA juga melibatkan pengendalian proses produksi untuk memastikan bahwa produk atau jasa yang dihasilkan sesuai dengan spesifikasi. Tugas utama QA *Tester/Engineering* adalah melakukan pengujian terhadap perangkat, membuat alur pengujian, serta membuat laporan hasil pengujian. Seorang QA tidak dapat bekerja sendiri, melainkan harus berkomunikasi dengan pihak-pihak yang berkepentingan seperti pengembang UI/UX, *Back-End* atau *Project Manager*.



Gambar 2. 4 Diagram Alir Proses SQA

Berdasarkan gambar 2.4 terdapat beberapa tahapan dalam proses *quality assurance*, yaitu sebagai berikut :

1. *Ticket Delivered* : Tahap ini dimulai ketika tim *Customer Support* melaporkan bug atau masalah dalam perangkat lunak. Ataupun seorang *Project Manager* maupun *Product Owner* membuat permintaan *enhancement* atau fitur baru untuk perangkat lunak. Laporan ini biasanya disampaikan dalam bentuk tiket atau permintaan dukungan. Tiket ini kemudian diserahkan kepada tim QA untuk dianalisis lebih lanjut.
2. *Analyze / Repro & Test Case* : Setelah menerima tiket bug, tim QA akan menganalisis masalah yang dilaporkan dan mencoba untuk mereproduksi *bug* tersebut atau menganalisis *requirement* dari permintaan *enhancement* atau fitur baru. Selama proses ini, akan dikembangkan test case atau rencana pengujian

yang diperlukan untuk memvalidasi perbaikan bug ataupun enhancement/fitur baru.

3. *Fixing / Development* : Begitu bug berhasil direproduksi atau analisis *requirement* selesai dan test case sudah disiapkan, tiket akan dialokasikan kepada tim pengembang untuk memperbaiki masalah tersebut. Pengembang akan bekerja untuk mengidentifikasi dan memperbaiki bug ataupun mengembangkan enhancement atau fitur dalam kode perangkat lunak.
4. *Testing* : Setelah perbaikan *bug* diselesaikan oleh tim pengembang, tahap pengujian dimulai kembali oleh tim QA. Mereka akan menguji perbaikan yang diimplementasikan untuk memastikan bahwa bug telah diperbaiki dengan benar atau *enhancement/fitur* berjalan dengan baik dan tidak menyebabkan dampak negatif pada fungsionalitas lainnya.
5. *Deployment* : Jika perbaikan telah lolos uji dan dianggap aman untuk digunakan, hasil perbaikan akan didistribusikan ke lingkungan produksi. Ini melibatkan proses implementasi perubahan ke dalam lingkungan produksi sehingga pengguna akhir dapat mengaksesnya.
6. *Test after Release* : Meskipun perangkat lunak telah di-*deploy*, proses SQA tidak berakhir di sini. Tim QA akan terus memantau perangkat lunak yang telah dirilis untuk memastikan tidak ada bug baru yang muncul atau dampak negatif lainnya. Jika ditemukan masalah setelah rilis, siklus pengujian dan perbaikan akan diulangi.
7. *Ticket Done* : Tahap ini menandai penyelesaian tiket atau permintaan *enhancement/fitur*. Setelah semua langkah dalam proses SQA telah diselesaikan dan perangkat lunak telah diperbaiki atau ditingkatkan sesuai dengan kebutuhan, tiket tersebut dianggap selesai. Ini menutup siklus penanganan masalah atau permintaan fitur tersebut.

Quality Assurance biasanya difokuskan pada proses yang tepat untuk memberikan keyakinan kualitas dari sebuah produk yang dibuat. Produk yang dibuat umumnya sesuai dengan semua fungsi yang ada dalam produk requirement untuk mencegah terjadinya kecacatan aplikasi/produk. Hal ini dilakukan untuk memastikan bahwa produk yang dihasilkan memenuhi standar kualitas yang telah ditetapkan.

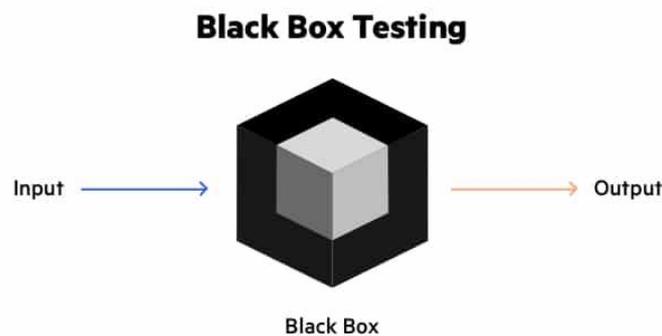
Proses Software Quality Assurance (SQA) dan metode penelitian yang digunakan memiliki beberapa tahapan yang berkesinambungan dan saling terkait untuk memastikan kualitas aplikasi Omnichannel Jubelio, yaitu :

1. **Studi Literatur dan Requirement Analysis:** Tahapan awal SQA, yaitu 'Ticket Delivered' dan 'Analyze/Repro & Test Case', berkaitan dengan studi literatur dan analisis kebutuhan dalam metode penelitian. Keduanya fokus pada pengumpulan informasi dan pemahaman kebutuhan sistem. Dalam SQA, tiket yang diserahkan oleh tim customer support atau manajer proyek memberikan informasi mengenai bug atau fitur yang perlu dianalisis, serupa dengan bagaimana penelitian mengumpulkan informasi dari berbagai sumber.
2. **Test Planning dan Test Case Development:** Sama seperti 'Fixing/Development' dan 'Testing' dalam SQA, tahapan test planning dan pengembangan test case dalam metode penelitian melibatkan perencanaan strategi pengujian dan pengembangan skenario pengujian untuk memastikan semua kebutuhan aplikasi tercakup. Di sini, QA mengembangkan rencana pengujian dan skenario yang diperlukan untuk memvalidasi perbaikan atau pengembangan fitur baru.
3. **Environment Setup dan Test Execution:** Tahapan 'Deployment' dalam SQA berkaitan erat dengan setup lingkungan dan eksekusi pengujian dalam metode penelitian. Pada tahap ini, QA menyiapkan lingkungan pengujian dan menguji perbaikan yang telah diimplementasikan oleh tim pengembang, serupa dengan bagaimana lingkungan pengujian disiapkan dan kasus pengujian dijalankan dalam penelitian Anda.
4. **Test Cycle Closure:** Tahap terakhir dari SQA, yaitu '*Test after Release*' dan '*Ticket Done*', berkaitan dengan penutupan siklus pengujian dalam metode penelitian. Setelah pengujian selesai, QA melakukan analisis terhadap hasil pengujian untuk memastikan tidak ada bug baru dan bahwa sistem berjalan sesuai spesifikasi. Ini mirip dengan analisis hasil pengujian dan perhitungan efektivitas dalam penelitian.

Secara keseluruhan, proses SQA dan metode penelitian yang digunakan merupakan proses yang selaras dalam memastikan kualitas dan keandalan aplikasi Omnichannel Jubelio. Keduanya fokus pada pengumpulan informasi, perencanaan dan pengembangan pengujian, eksekusi pengujian, serta analisis hasil untuk memastikan aplikasi memenuhi standar kualitas yang ditetapkan.

2.4 *Blackbox Testing*

Black Box Testing merupakan metode pengujian perangkat lunak yang dilakukan tanpa mengetahui bagaimana kode program di dalam sistem tersebut [3]. Tujuan dari pengujian ini adalah untuk mengetahui apakah sistem yang diuji sudah sesuai dengan spesifikasi yang telah ditentukan atau tidak.



Gambar 2. 5 Konsep *Black Box Testing*

Pada *Black Box Testing*, pengujian dilakukan pada detail aplikasi seperti *user interface* aplikasi, fungsi atau fitur dalam aplikasi, dan kesesuaian alur aplikasi dengan bisnis proses yang diinginkan oleh pengguna. Seorang *Quality Assurance* harus mampu membuat *test case* untuk melakukan *black box testing* dengan menguji fitur-fitur pada aplikasi, memastikan *flow* fitur sesuai dengan *user requirement*, dan menemukan *bug* atau eror pada tampilan aplikasi. Tahun 2022 menjadi saksi penelitian Aulizar Arfan tentang Software Quality Assurance menggunakan Metode *black box*, dengan mengambil studi kasus di PT. GIT Solution. Hasil dari penelitian ini menunjukkan pengujian *black box* dengan

menjalankan pengujian manual maupun otomatis dapat menemukan *bug* pada fungsionalitas dalam aplikasi dengan baik [11]. Dalam hal ini, *Black box Testing* dapat membantu dalam memastikan bahwa produk atau layanan yang dihasilkan sesuai dengan standar kualitas yang telah ditentukan oleh *Quality Assurance*. Berikut terdapat 10 teknik pengujian dari metode Black Box [12]:

1. *Equivalence Partitioning*

Pengujian yang dilakukan dengan membagi menjadi beberapa kelas partisi berbeda untuk mengurangi kasus uji.

2. *Boundary Value Analysis*

Pengujian ini dilakukan pada batasan nilai input. BVA melengkapi pengujian EP.

3. *Comparison Testing*

Pengujian setiap versi dengan data yang sama untuk menguji semua versi menampilkan keluaran yang sama.

4. *Sample Testing*

Pengujian yang menggunakan beberapa nilai yang terpilih dari sebuah kelas ekuivalen, bukan menguji sebuah data.

5. *Robustness Testing*

Pengujian ini untuk membuktikan bagaimana sistem bereaksi terhadap input yang tidak valid atau tak terduga.

6. *Behaviour Testing*

Pengujian dilakukan dengan cara membuat data baru secara berulang untuk menghindari *stack*.

7. *Requirement Testing*

Pengujian ini memastikan bahwa semua kebutuhan dan persyaratan yang telah ditentukan sudah dipenuhi oleh sistem.

8. *Performance Testing*

Pengujian yang dilakukan untuk mengevaluasi kemampuan program dalam bekerja dengan kebutuhan lain seperti : aliran data, ukuran pemakaian memori, kecepatan *load*.

9. *Endurance Testing*

Pengujian dengan kasus uji yang diulang-ulang dengan jumlah tertentu untuk memeriksa apakah sistem sudah sesuai dengan spesifikasi.

10. Cause Effect Graph Testing

Pengujian yang menggunakan kondisi input (cause) dan kondisi output (effect).

2.5 Test Case

Test Case adalah kumpulan skenario yang dibuat oleh tim QA untuk memastikan bahwa sistem yang akan diuji memenuhi persyaratan dan standar tertentu serta berfungsi dengan baik[13]. *Test Case* ini bertujuan untuk memastikan bahwa sistem dapat dijalankan dengan baik dan sesuai dengan kebutuhan awal, serta mampu memberikan respon terhadap masukan yang valid maupun tidak valid.

Berikut adalah komponen-komponen yang ada dalam *Test Case*:

- *Test Case ID* : Nama atau identifikasi dari *Test Case* yang memudahkan penguji dalam melacak error atau mengidentifikasi persyaratan perangkat lunak pada tahap berikutnya.
- *Test Case Description* : Deskripsi singkat tentang apa yang akan diuji dalam *Test Case* tersebut.
- *Pre-condition* : Kondisi awal yang harus terpenuhi sebelum *Test Case* dapat dijalankan.
- *Test Step* : Langkah-langkah konkret yang harus diikuti oleh penguji untuk menguji suatu fitur atau fungsi.
- *Expected Result*: Hasil yang diharapkan setelah menjalankan langkah-langkah pengujian.
- *Actual Result* : Hasil yang sebenarnya setelah menjalankan langkah-langkah pengujian.
- Status : Status pengujian, apakah *Test Case* tersebut berhasil (*pass*) atau gagal (*fail*).

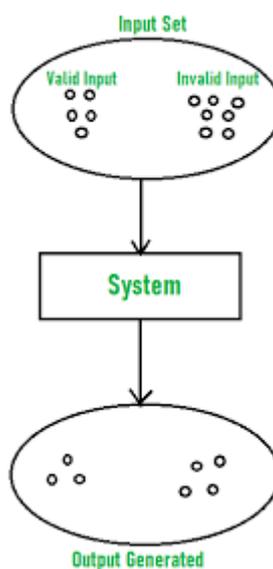
Test Case memiliki beberapa kegunaan, antara lain:

- Sebagai panduan : *Test Case* digunakan sebagai panduan bagi penguji dalam menjalankan pengujian dengan langkah-langkah yang telah ditentukan.
- Sebagai dokumentasi : *Test Case* juga berfungsi sebagai dokumentasi yang mencatat hasil pengujian, baik yang berhasil maupun yang gagal.
- Sebagai alat pelaporan : Hasil dari pengujian *Test Case* dapat digunakan untuk melaporkan keadaan sistem kepada tim pengembang atau pemangku kepentingan lainnya.

Test case yang baik adalah *test case* yang mempunyai kemungkinan tinggi dalam menemukan kesalahan dan cacat yang sebelumnya belum ditemukan, bukan yang dapat memperlihatkan bahwa program telah bekerja dengan benar [14].

2.6 Equivalence Partitioning

Metode *Equivalence Partitioning* adalah metode pengujian berdasarkan nilai *input* suatu *form* pada setiap menu aplikasi web yang akan diuji. Setiap nilai yang diinput akan diuji dan disusun berdasarkan fungsinya, yaitu apakah itu valid atau tidak valid[15].



Gambar 2. 6 Ilustrasi Metode *Equivalence Partitioning*

Pada gambar 2.6 mengilustrasikan metode *Equivalence Partitioning* (EP) dalam pengujian perangkat lunak. Metode ini membagi input yang mungkin ke dalam beberapa partisi atau kelas yang dianggap memiliki perilaku yang sama. Di bagian atas gambar, terdapat dua kelompok utama dalam kumpulan input: Valid Input (input yang sesuai dengan persyaratan dan menghasilkan output yang benar) dan Invalid Input (input yang tidak sesuai dengan persyaratan dan menghasilkan output kesalahan). Masing-masing kelompok ini diwakili oleh beberapa lingkaran yang menunjukkan variasi input dalam partisi tersebut. Lingkaran-lingkaran di bagian output menunjukkan berbagai hasil yang mungkin dihasilkan oleh sistem setelah memproses input. Di sisi kiri, terdapat 3 lingkaran yang mewakili output yang dihasilkan dari input valid, sedangkan di sisi kanan, terdapat 4 lingkaran yang mewakili output atau pesan kesalahan yang dihasilkan dari input tidak valid.

Jumlah lingkaran pada output menunjukkan variasi hasil yang dapat dihasilkan oleh sistem tergantung validitas input yang diterima. Pada ilustrasi terdapat 5 valid input yang digambarkan menjadi 5 lingkaran, saat diproses oleh sistem hanya menyisakan 3 lingkaran. Begitu pula dengan 7 invalid input yang diproses oleh sistem hanya menyisakan 3 lingkaran output. Hal tersebut terjadi karena beberapa hal. Yang pertama, karena adanya kombinasi input dan pengelompokan. Misal, input A dan input B keduanya valid dan menghasilkan satu output yang sama (output 1) atau bisa juga input B, C, dan D dapat menghasilkan output gabungan (output 2 dan output 3). Atau hal lain, seperti *filtering* atau pengabaian, dimana sistem mungkin menyaring sehingga terdapat beberapa valid input yang mungkin tidak perlu ditampilkan dalam output akhir.

Equivalence Partitioning (EP) digunakan dalam pengujian perangkat lunak untuk membagi berbagai jenis input menjadi beberapa kelompok yang dianggap sama dalam cara mereka diproses oleh sistem. Dengan menguji minimal satu input dari setiap kelompok, dapat dipastikan bahwa sistem memproses semua jenis input dengan benar. Ini membantu mengurangi jumlah pengujian yang perlu dilakukan, karena tidak perlu menguji setiap input secara individu. Sebaliknya, hanya perlu menguji beberapa perwakilan dari setiap kelompok, sehingga tetap dapat memastikan bahwa semua fungsi aplikasi bekerja dengan benar tanpa harus melakukan terlalu banyak pengujian.

Berikut pedoman dan cara pengujian *Equivalence Partitioning* [16]

1. Jika kondisi rentang diberikan sebagai input, maka satu kelas ekuivalen yang valid dan dua yang tidak valid ditentukan.

Contoh :

- Input: Usia
- Rentang Valid: 18 hingga 60 tahun
- Kelas Partisi :
 - Valid: 18-60 tahun (contoh: 25, 45)
 - Tidak Valid: <18 tahun (contoh: 15) dan >60 tahun (contoh: 65)

2. Jika nilai tertentu atau spesifik diberikan sebagai input, maka satu kelas ekuivalen yang valid dan dua yang tidak valid ditentukan.

Contoh:

- Input: Bulan dalam setahun
- Nilai Valid: Januari hingga Desember
- Kelas Partisi :
 - Valid: Januari hingga Desember (contoh: Januari, Maret)
 - Tidak Valid: Nilai di luar bulan (contoh: 13, -1) dan nilai bukan bulan (contoh: "ABC", 0.5)

3. Jika anggota himpunan diberikan sebagai input, maka satu kelas ekuivalen yang valid dan satu yang tidak valid ditentukan.

Contoh:

- Input: Warna (hanya warna dasar)
- Himpunan Valid: Merah, Biru, Hijau
- Kelas Partisi :
 - Valid: Merah, Biru, Hijau (contoh: Merah)
 - Tidak Valid: Warna lain (contoh: Ungu, Hitam)

4. Jika *boolean* diberikan sebagai kondisi input, maka satu kelas ekuivalen valid dan satu kelas invalid ditentukan.

Contoh:

- Input: Apakah pengguna setuju dengan syarat dan ketentuan?
- Nilai Valid: True (setuju)

- Kelas Partisi :
 - Valid: True (contoh: setuju)
 - Tidak Valid: False (contoh: tidak setuju)
- 5. Tentukan kriteria dan partisi sesuai dengan pedoman di atas
- 6. Lakukan pengujian pada data uji dan hitung Tingkat keberhasilan aplikasi dengan rumus :

$$\text{Nilai Efektivitas} = \left(\frac{\sum \text{Pengujian Berhasil}}{\sum \text{Skenario Pengujian}} \right) \times 100\%$$

Terdapat beberapa kelebihan dan kekurangan pada metode *Equivalence Partitioning*, yaitu sebagai berikut [17] :

Kelebihan EP:

- Memberikan sensasi pengujian yang menyeluruh tanpa perlu melakukan pengujian yang melelahkan
- Memungkinkan pengujian banyak jenis input atau output hanya dengan beberapa contoh dari setiap kelompok
- Menghindari pengujian berulang dengan memilih beberapa contoh dari setiap kelompok input.

Kekurangan EP :

- Hanya cocok untuk data input dengan rentang dan nilai tertentu.
- Mengasumsikan bahwa semua data dalam kelompok yang sama diproses dengan cara yang sama oleh sistem.
- Tidak bisa menangani kesalahan di batas nilai, perlu dikombinasikan dengan *Boundary Value Analysis*.

2.7 *Boundary Value Analysis*

Metode *Boundary Value Analysis* adalah metode pengujian berdasarkan suatu *range* dengan menentukan batas atas dan batas bawah dari nilai *input* suatu form [4] Metode ini digunakan untuk memastikan bahwa data *input* yang melebihi batas yang ditentukan tidak dapat disimpan dengan benar dalam *database* [18]. Contoh

dari penerapan pengujian menggunakan *boundary value analysis* adalah nilai yang digunakan adalah nilai minimal-1, minimal, minimal+1, maksimum-1, maksimum, dan maksimum+1. Untuk batasan data nilai *input* BVA dapat dilihat pada tabel 2.1 berikut :

Tabel 2. 1 Batasan Data *Boundary Value Analysis*

Kategori	Keterangan
Min	Nilai minimum
Min-1	Tepat berada di bawah nilai minimum
Min+1	Tepat berada di atas minimum
Max	Nilai maksimum
Max-1	Tepat berada di bawah nilai maksimum
Max+1	Tepat berada di atas maksimum

Pada metode ini diharuskan membuat kasus uji untuk menguji program di batas-batas tersebut, serta nilai-nilai tepat di dalam dan di luar batas-batas tersebut. Metode ini hanya dapat menguji tipe data yang memiliki nilai rentang dan nilai yang jelas. Setelah mengetahui batasan data pada metode BVA, berikut pedoman dan cara pengujian menggunakan metode *Boundary Value Analysis* [19] :

1. Jika kondisi input pada rentang dibatasi dengan nilai a dan b, uji dengan nilai (a-1), a, (a+1), (b-1), b, (b+1).

Contoh:

- Kasus : Validasi Panjang username
- Rentang:
 - a (minimum) : 5 karakter
 - b (maksimum) : 10 karakter
- Pengujian:
 - (a-1): 4 karakter (tidak valid)
 - a: 5 karakter (valid)

- (a+1): 6 karakter (valid)
 - (b-1): 9 karakter (valid)
 - b: 10 karakter (valid)
 - (b+1): 11 karakter (tidak valid)
2. Jika input menggunakan sejumlah n nilai, uji dengan (n-1), n, dan (n+1) nilai.

Contoh:

- Input: Jumlah item yang harus dibeli untuk mendapatkan diskon
 - Jumlah: n = 5
 - Pengujian:
 - (n-1) = 4 (tidak mendapatkan diskon)
 - n = 5 (mendapatkan diskon)
 - (n+1) = 6 (mendapatkan diskon)
3. Lakukan 2 langkah sebelumnya untuk memproses output
4. Jika program memiliki batasan internal seperti ukuran buffer atau batas array, gunakan data input yang menguji batasan tersebut.
5. Buat tabel pengujian hasil untuk nilai maksimal dan minimal, dan hitung tingkat keberhasilan aplikasi dengan rumus :

$$\text{Nilai Efektivitas} = \left(\frac{\sum \text{Pengujian Berhasil}}{\sum \text{Skenario Pengujian}} \right) \times 100\%$$

Terdapat beberapa kelebihan dan kekurangan pada metode *Bounday Value Analysis*, yaitu sebagai berikut [17] :

Kelebihan BVA:

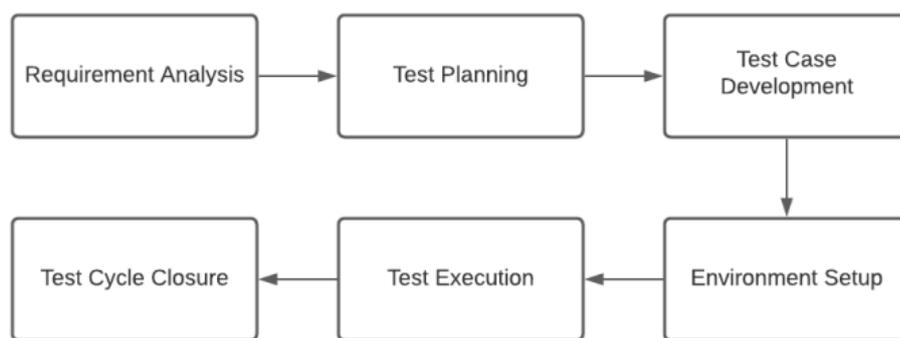
- Melengkapi pengujian *Equivalence Partitioning* dengan menangani kesalahan di batas kelas,
- Cocok untuk variabel yang memiliki jumlah fisik tertentu

Kekurangan BVA:

- Menghasilkan jumlah kasus uji yang tinggi.
- Tidak dapat digunakan untuk variabel *Boolean* dan logis.

2.8 Software Testing Life Cycle (STLC)

Software Testing Life Cycle (STLC) merupakan salah satu siklus tahapan kegiatan dalam proses pengujian pada suatu perangkat lunak untuk mengetahui kualitas daripada produk perangkat lunak itu sendiri [20]. Proses pengujian ini mempunyai tahapan–tahapan yang spesifik.



Gambar 2. 7 Tahapan *Software Testing Life Cycle* (STLC)

Berdasarkan gambar 2.7, terdapat 6 tahap dalam STLC, yaitu sebagai berikut :

1. *Requirement Analysis* : Tahap ini dilakukan untuk memahami kebutuhan pengguna terkait desain perangkat lunak dan menentukan kriteria pengujian yang sesuai. Pada tahap ini, dilakukan analisis terhadap dokumen-dokumen kebutuhan dan spesifikasi perangkat lunak.
2. *Test Planning* : Tahap ini dilakukan untuk merencanakan strategi pengujian, menentukan lingkup pengujian, jenis pengujian, dan jadwal pengujian.
3. *Test Case Development* : Tahap ini dilakukan untuk membuat skenario pengujian yang terstruktur dan terdokumentasi dengan baik. Pada tahap ini, dilakukan pembuatan skenario pengujian yang mencakup langkah-langkah pengujian, data pengujian, dan hasil yang diharapkan berdasarkan kebutuhan desain perangkat lunak.
4. *Test Environment Setup* : Tahap ini dilakukan untuk menyiapkan lingkungan

pengujian yang sesuai. Pada tahap ini, biasa dilakukan instalasi perangkat lunak, konfigurasi lingkungan pengujian, dan persiapan data pengujian.

5. *Test Execution* : Tahap ini dilakukan untuk menjalankan skenario pengujian dan mengumpulkan hasil pengujian ke dalam laporan hasil uji.
6. *Test Cycle Closure* : Tahap ini dilakukan untuk mengevaluasi hasil pengujian dan menyusun laporan pengujian. Pada tahap ini, dilakukan evaluasi terhadap hasil pengujian, penyusunan laporan pengujian, dan identifikasi tindakan perbaikan yang perlu dilakukan.

Dalam prosesnya, kegiatan dilakukan dan dirancang secara sistematis dan terencana.

2.9 Penelitian Terkait

Pada penyusunan skripsi ini terdapat lima penelitian terkait yang akan digunakan sebagai referensi penelitian, yang pertama ada Disa dan Agus dalam penelitiannya pada Aplikasi Perhitungan Stok Barang pada CV. Delta Pilar [21]. Pada penelitiannya Disa dan Agus melakukan pengujian *black box* dengan 2 teknik yaitu campuran teknik *Equivalence Partitioning* dengan *Boundary Value Analysis* dan teknik *Decision Table* pada form yang terdapat dalam aplikasi tersebut. Hasilnya menunjukkan bahwa form dalam aplikasi mampu untuk memvalidasi inputan kosong, tetapi tidak mampu memvalidasi jenis inputan invalid. Maka, disarankan untuk pihak Perusahaan untuk meningkatkan kualitas form dengan memvalidasi jumlah dan jenis karakter.

Pengujian *Black Box* dengan *Equivalence Partitioning* dan *Boundary Value Analysis* juga dilakukan oleh Putu dkk dan Fairus dkk. Putu melakukan pengujian pada Sistem Informasi Akademik Universitas Mataram dan berhasil menemukan *defect* dengan kategori invalid pada fitur pada aplikasi tersebut. Pengujian dilakukan sebanyak 322 *test case* dengan 89 *defect* pada fitur *log in*, *HTTP ERROR 500*, serta ketidaksesuaian GUI dengan inputan [22]. Sedangkan Fairus dkk menguji aplikasi yang berbeda yaitu aplikasi *e-commerce* pada fitur *checkout* [23]. Adapun perbedaan hasil persentase pada pengujian menggunakan metode *Boundary Value Analysis* yaitu 100%, sedangkan dengan metode *Equivalence*

Partitioning berjumlah 57,14% yang berarti masih terdapat *defect* dalam aplikasi tersebut.

Tidak harus menggunakan kedua metode EP dan BVA, banyak peneliti yang menggunakan hanya satu metode, salah satunya metode *Equivalence Partitioning*. Marfuah dan Stefi menggunakan metode EP untuk menguji Sistem Pemberdayaan Pemerataan UMKM dan tidak menemukan adanya kesalahan pada fungsionalitas form. Hal ini menunjukkan aplikasi berkualitas baik [24]. Begitupun dengan Ilham dkk yang tidak menemukan kesalahan pada fungsionalitas aplikasi *e-commerce* berbasis *website*. Pada penelitian Ilham dkk, 15 test case yang diuji berhasil berjalan dengan baik [25].

Tabel 2. 2 Penelitian Terkait

No.	Peneliti	Objek Penelitian	Metode	Hasil
1.	Disa Arista Syuhada dan Agus Nursikuwagus [21]. (2019)	Aplikasi Perhitungan Stok Barang CV. Delta Pilar	Black Box Testing	<ul style="list-style-type: none"> Defect dari teknik BVA + EP = 43,4% Defect dari teknik DT = 53,5%
2.	Putu Ayu Desi Anggara Santi, Royana Afwani, Moh. Ali Albar [22]	Sistem Informasi Akademik Universitas Mataram	Blackbox Testing, Equivalence Partitioning, Boundary Value Analysis	<ul style="list-style-type: none"> Defect = 80 Passed = 75.16% Failed = 24.84%
3.	Fairus In'am Pratama, Eka Mira Novita Subroto, Radina Mutia	Aplikasi E-Commerce OpenCart	Black Box Testing, Equivalence Partitioning,	<ul style="list-style-type: none"> Hasil pengujian EP = 57,14% Hasil pengujian BVA = 100%

	Haira, Muhammad Ainul Yaqin [23]		Boundary Value Analysis	
4.	Marfuah dan Steffi Adam[24]	Sistem Pemberdayaan Pemerataan UMKM	Black Box Testing, Equivalence Partitioning	Defect = 0
5.	Muhammad Ilham, David Apriansyah, Muhammad Rendy Nurhakiki, Novi Apriani, Aries Saifudin [25]	Aplikasi Ecommerce Berbasis Website	Black Box Testing, Equivalence Partitoning	Defect = 0

Pada penelitian ini akan dilakukan pengujian pada aplikasi dengan industri berbeda yaitu aplikasi *omnichannel*. Perbedaan penelitian ini dengan penelitian terkait di atas yaitu objek penelitian yang merupakan *omnichannel*. Saat ini belum ada penelitian pengujian terkait pada aplikasi jenis tersebut. Pengujian ini akan menggunakan metode black box dengan *teknik equivalence partitioning* dan *boundary value analysis* pada fitur Pemrosesan Pesanan dalam modul Gudang

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilakukan di PT Guardia Teknologi (Jubelio) yang berada di Sampoerna Strategic Square North Tower, Jl. Jenderal Sudirman No.Kav. 45-46 Floor 16, Karet Semanggi, Kec. Setiabudi, Kota Jakarta Selatan, Jakarta . Penelitian ini dilaksanakan mulai bulan Januari 2024 sampai bulan Juli 2024.

Tabel 3. 1 Jadwal Penelitian

No.	Aktivitas	Januari 2024	Februari 2024	Maret 2024	April 2024	Mei 2024	Juni 2024	Juli 2024
1.	Studi Literatur							
2.	Requirement Analysis							
3.	Test Planning							
4.	Test Case Development							
5.	Environment Setup							
6.	Test Execution							
7.	Test Cycle Closure (Pembuatan Laporan)							

3.2 Alat dan Bahan

1. Alat Penelitian

Dalam penelitian ini digunakan beberapa alat yang terdiri dari perangkat keras dan perangkat lunak sebagai berikut :

a. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian adalah salah satu unit laptop HP ENVY X360 dengan spesifikasi *processor* AMD *Ryzen* 5 dan memori RAM 8 GB.

b. Perangkat Lunak

Perangkat lunak yang menunjang penelitian ini adalah sistem operasi *Windows* 10, *Microsoft Office* 2019, *Mendeley Reference Manager*, serta web browser *Google Chrome*

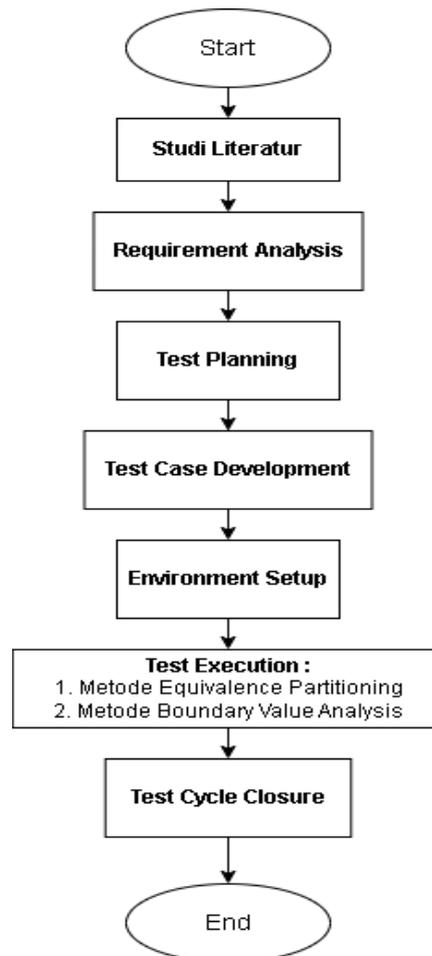
2. Bahan Penelitian

Berbagai bahan dan informasi yang digunakan dalam penelitian ini yaitu sebagai berikut :

- a. Berbagai literatur dari jurnal dan beberapa penelitian terkait.
- b. Website *Omnichannel Jubelio*

3.3 Tahapan Penelitian

Penelitian mengenai pengujian *black box* dengan metode *equivalence partitioning* dan *boundary value analysis* pada *Omnichannel Jubelio* mengimplementasi tahapan *Software Testing Life Cycle* (STLC) yang dapat dilakukan dalam beberapa tahapan seperti pada Gambar 3.1 :



Gambar 3. 1 Tahapan Penelitian

1. Studi Literatur

Pada tahapan ini dilakukan dengan mengumpulkan data dan informasi dari berbagai buku elektronik, jurnal, dan berbagai bacaan yang berkaitan dengan pengujian *black-box*, metode *equivalence partitioning*, *boundary value analysis*, dan strategi *omnichannel*, termasuk studi kasus yang relevan dan penelitian terdahulu.

2. Requirement Analysis

Pada tahapan ini dilakukan dengan menganalisis kebutuhan aplikasi dengan mengumpulkan spesifikasi dan dokumen teknis yang berkaitan dengan fitur

Warehouse Management System (WMS) khususnya di fitur pesanan pada modul Penjualan dan Gudang. Analisis kebutuhan ini juga dapat dilakukan dengan berdiskusi dengan stakeholder, tim pengembang ataupun manajemen di PT Guardia Teknologi (Jubelio) untuk memahami fitur dan fungsionalitas dari aplikasi. Berikut adalah hasil analisis karakteristik Input pada aplikasi Omnichannel Jubelio berdasarkan spesifikasi :

1. **Input Berupa Rentang Nilai**

Banyak *field* input di aplikasi ini yang memiliki rentang nilai, seperti jumlah pesanan atau tanggal. Pengujian nilai batas dari rentang ini (dengan BVA) memastikan bahwa sistem menangani nilai ekstrem dengan benar.

2. **Input Kategorikal**

Beberapa input merupakan kategori yang terbatas, seperti jenis kurir atau status pengiriman. EP efektif untuk menguji input jenis ini dengan mengelompokkan kategori-kategori tersebut ke dalam partisi yang ekuivalen dan menguji representasi dari setiap partisi.

3. **Validasi Data**

Form di aplikasi ini sering kali membutuhkan validasi data yang ketat, seperti validitas email dan format nomor telepon. EP dan BVA dapat digunakan untuk memastikan bahwa validasi ini berfungsi dengan benar dengan menguji berbagai skenario input yang valid dan tidak valid.

Pemilihan metode *Equivalence Partitioning* (EP) dan *Boundary Value Analysis* (BVA) untuk pengujian aplikasi Omnichannel Jubelio didasarkan pada karakteristik input aplikasi yang beragam dan terstruktur. Metode EP memungkinkan pengelompokan input menjadi beberapa kategori yang setara untuk pengujian yang lebih fokus dan lebih efisien karena hanya satu atau beberapa nilai dari setiap partisi yang diuji. Sedangkan metode BVA memastikan bahwa sistem dapat menangani nilai-nilai ekstrem di batas-batas input dengan baik. Kombinasi kedua metode ini memberikan cakupan pengujian yang luas dan mendetail, memastikan kualitas dari form pada aplikasi dengan pengujian yang efisien dan efektif.

3. Test Planning

Setelah dilakukan analisis kebutuhan, maka akan dilakukan strategi pengujian berdasarkan hasil analisis kebutuhan, studi literatur, dan dokumen teknis aplikasi. Tahap ini dilakukan untuk merencanakan strategi pengujian, menentukan lingkup pengujian, metode pengujian, dan jadwal pengujian.

4. Test Case Development

Tahap ini dilakukan untuk membuat skenario pengujian yang terstruktur dan terdokumentasi dengan baik. Pada tahap ini akan dikembangkan kasus pengujian atau test case yang mencakup skenario pengujian untuk fitur-fitur sistem yang mencakup langkah-langkah pengujian, data pengujian, dengan menetapkan data *input* dan hasil yang diharapkan berdasarkan kebutuhan desain perangkat lunak. Berikut ini merupakan beberapa skenario pengujian dari fitur proses pesanan pada aplikasi *omnichannel* Jubelio :

Tabel 3. 2 Skenario Pengujian - Login User

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC01	Login dengan username dan password yang benar	EP	Valid	Berhasil login dan masuk ke halaman utama
TC02	Login dengan username dan password huruf kapital semua	EP	Invalid	Gagal login dan muncul pesan eror " <i>Invalid email or password</i> "
TC03	Login dengan username dan password yang tidak terdaftar	EP	Invalid	Gagal login dan muncul pesan eror " <i>Invalid email or password</i> "
TC04	Login dengan username benar dan password yang salah	EP	Invalid	Gagal login dan muncul pesan eror " <i>Invalid email or password</i> "

TC05	Login dengan password yang benar dan username salah	EP	Invalid	Gagal login dan muncul pesan eror " <i>Invalid email or password</i> "
TC06	Login dengan username dan password kosong	BVA	Min-1	Gagal login dan muncul pesan eror " <i>Tidak boleh kosong</i> "
TC07	Login dengan username benar dan password kosong	BVA	Min-1	Gagal login dan muncul pesan eror " <i>Tidak boleh kosong</i> "
TC08	Login dengan password benar dan username kosong	BVA	Min-1	Gagal login dan muncul pesan eror " <i>Tidak boleh kosong</i> "

Pada tabel 3.2 terdapat tabel skenario pengujian yang berfokus pada aktivitas login user. Dalam tabel tersebut terdapat delapan skenario pengujian, dengan lima skenario menggunakan metode EP dan tiga skenario menggunakan metode BVA. Setiap skenario diuji dengan variasi input yang berbeda untuk memastikan bahwa sistem dapat menangani berbagai kondisi login dengan benar.

TC01 hingga TC05 menggunakan metode EP untuk menguji skenario login dengan variasi input yang valid dan invalid. TC01 memeriksa login dengan username dan password yang benar (valid), yang diharapkan berhasil dan masuk ke halaman utama. TC02 hingga TC05 menguji input yang invalid, seperti menggunakan huruf kapital semua, username dan password yang tidak terdaftar, kombinasi username benar dan password salah, serta kombinasi password benar dan username salah. Semua skenario invalid ini diharapkan gagal login dan menampilkan pesan kesalahan "Invalid email or password".

TC06 hingga TC08 menggunakan metode BVA untuk menguji skenario dengan input batas minimum (Min-1). TC06 memeriksa kondisi apakah username dan password keduanya kosong, TC07 menguji username benar tetapi password kosong, dan TC08 menguji password benar tetapi username kosong. Semua skenario ini diharapkan gagal login dan menampilkan pesan kesalahan "Tidak boleh kosong".

Tabel 3. 3 Skenario Pengujian - Cetak Picklist

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC09	Klik button Cetak Picklist dengan memilih pesanan terlebih dahulu	EP	Valid	Berhasil masuk ke pop up Cetak Picklist
TC10	Klik button Cetak Picklist tanpa memilih pesanan	EP	Invalid	Tidak ada button Cetak Picklist
TC11	Scan id picker dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama picker.
TC12	Scan id picker dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC13	Scan id picker dengan mengosongkan inputan	BVA	Min-1	Sistem menampilkan error message <i>“Tidak boleh kosong”</i>
TC14	Scan id picker dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC15	Scan id picker dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>

Pada tabel 3.3 merupakan tabel skenario pengujian fitur Cetak Picklist yang terdapat pada Tab Picking. Terdapat tujuh skenario pengujian pada fitur tersebut yang berfokus pada form input dengan enam skenario menggunakan metode EP dan satu skenario menggunakan metode BVA. TC09 dan TC10 menguji fungsi Cetak Picklist dengan dan tanpa memilih pesanan terlebih dahulu menggunakan metode EP. TC09 adalah skenario positif dengan cara pengguna memilih pesanan terlebih dahulu, diharapkan berhasil menampilkan pop-up Cetak Picklist. TC10 adalah skenario negatif dengan pengguna tidak memilih pesanan, diharapkan sistem tidak menampilkan tombol Cetak Picklist.

TC11 dan TC12 menguji pemindaian ID picker dengan input ID yang valid dan tidak valid menggunakan metode EP. TC11 adalah skenario positif jika ID picker terdaftar, sistem diharapkan menerima input dan menampilkan nama picker. TC12 adalah skenario negatif jika ID picker tidak terdaftar, sistem diharapkan menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

TC13 menggunakan metode BVA untuk menguji skenario jika input ID picker kosong. Ini adalah skenario negatif jika sistem diharapkan menampilkan pesan kesalahan "Tidak boleh kosong".

TC14 dan TC15 menguji pemindaian ID picker dengan menambahkan spasi di depan dan di belakang menggunakan metode EP. Kedua skenario ini adalah skenario negatif jika sistem diharapkan menolak input dengan menampilkan pesan kesalahan "Data tidak ditemukan".

Tabel 3. 4 Skenario Pengujian - Serahkan ke Picker

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC16	Scan id picker dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama picker.
TC17	Scan id picker dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC18	Scan id picker dengan mengosongkan input	BVA	Min-1	Sistem menampilkan error message " <i>Tidak boleh kosong</i> "
TC19	Scan id picker dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC20	Scan id picker dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "

TC21	Scan nomor pesanan dengan pesanan yang terdapat di Picking – Belum Mulai	EP	Valid	Sistem menerima inputan dan berhasil membuat picklist
TC22	Scan nomor pesanan dengan pesanan yang tidak terdapat di Picking – Belum Mulai	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Pesanan sudah di-picking”</i>
TC23	Scan nomor pesanan dengan input salah	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC24	Scan nomor pesanan dengan mengosongkan inputan	BVA	Min-1	Sistem menolak inputan dan menampilkan error message <i>“Tidak boleh kosong”</i>
TC25	Scan no pesanan tanpa scan id picker terlebih dahulu	EP	Invalid	Sistem menampilkan error message <i>“Silahkan isi picker terlebih dahulu”</i>

Pada tabel 3.4 merupakan tabel skenario pengujian pada fitur Serahkan ke Picker di tab Picking. Terdapat 10 skenario pengujian dengan diantaranya delapan skenario menggunakan metode EP dan dua skenario menggunakan metode BVA. Pada TC16 dan TC17 menguji pemindaian ID picker dengan input ID yang valid dan tidak valid menggunakan metode EP. TC16 adalah skenario positif jika ID picker terdaftar, sehingga sistem diharapkan menerima input dan menampilkan nama picker. TC17 adalah skenario negatif saat ID picker tidak terdaftar, sehingga sistem diharapkan menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

TC18 menggunakan metode BVA untuk menguji skenario jika input ID picker kosong. Ini adalah skenario negatif jika sistem diharapkan menampilkan pesan kesalahan "Tidak boleh kosong".

TC19 dan TC20 menguji pemindaian ID picker dengan menambahkan spasi di depan dan di belakang menggunakan metode EP. Kedua skenario ini adalah skenario negatif jika sistem diharapkan menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

TC21 hingga TC23 menguji pemindaian nomor pesanan dengan input yang valid, tidak valid, dan salah menggunakan metode EP. TC21 adalah skenario positif jika nomor pesanan yang dipindai terdaftar di "Picking - Belum Mulai", sehingga sistem diharapkan menerima input dan berhasil membuat picklist. TC22 dan TC23 adalah skenario negatif jika nomor pesanan tidak terdaftar atau input salah, sehingga sistem diharapkan menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

TC24 menggunakan metode BVA untuk menguji skenario jika input nomor pesanan kosong. Ini adalah skenario negatif jika sistem diharapkan menampilkan pesan kesalahan "Tidak boleh kosong".

TC25 menguji skenario jika nomor pesanan dipindai tanpa memindai ID picker terlebih dahulu menggunakan metode EP. Ini adalah skenario negatif jika sistem diharapkan menampilkan pesan kesalahan "Silakan isi picker terlebih dahulu".

Tabel 3. 5 Skenario Pengujian - Mulai Packing

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC26	Scan id packer dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama picker.
TC27	Scan id packer dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC28	Scan id packer dengan mengosongkan input	BVA	Min-1	Sistem menampilkan error message " <i>Tidak boleh kosong</i> "
TC29	Scan id packer dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC30	Scan id packer dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "

TC31	Scan nomor pesanan yang terdapat di Packing – Belum Mulai	EP	Invalid	Sistem menerima inputan dan menampilkan detail pesanan
TC32	Scan nomor pesanan yang sudah pernah di-packing	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Pesanan sudah pernah dipacking”</i>
TC33	Scan nomor pesanan dengan input salah	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC34	Scan nomor pesanan tanpa scan id packer terlebih dahulu	EP	Invalid	Sistem menampilkan error message <i>“Silahkan isi packer terlebih dahulu”</i>
TC35	Scan nomor pesanan dengan mengosongkan input	BVA	Min-1	Sistem menampilkan error message <i>“Tidak boleh kosong”</i>
TC36	Input qty pack sesuai qty pesanan	BVA	n	Sistem menerima inputan dan menampilkan gambar produk di samping kanan
TC37	Input qty pack lebih besar dari qty pesanan	BVA	n+1	Sistem menolak inputan dan menampilkan error message <i>“Qty pack tidak boleh lebih besar dari qty pesanan”</i>
TC38	Input qty pack lebih kecil dari qty pesanan	BVA	n-1	Sistem menerima inputan dan menampilkan gambar produk di samping kanan
TC39	Input qty pack dengan mengosongkan input	BVA	Min-1	Sistem menolak inputan dan menampilkan error message <i>“Qty pack must be a number”</i>
TC40	Input qty pack dengan menambahkan spasi atau karakter lain	EP	Invalid	Sistem menolak inputan spasi atau karakter lain

Pada tabel 3.5 terdapat tabel skenario pengujian pada fitur Mulai Packing di tab Packing. Berdasarkan tabel, terdapat 15 skenario pengujian dengan sembilan skenario menggunakan metode EP dan enam skenario menggunakan metode BVA. Pada TC26 hingga TC30 menguji skenario pemindaian ID picker dengan input ID yang valid, tidak valid, kosong, dan dengan spasi tambahan menggunakan metode EP dan BVA. TC26 adalah skenario positif jika ID picker terdaftar, sistem diharapkan menerima input dan menampilkan nama picker. TC27 hingga TC30 adalah skenario negatif jika input ID tidak terdaftar, kosong, atau memiliki spasi tambahan, diharapkan sistem menolak input dan menampilkan pesan kesalahan.

TC31 hingga TC34 menguji skenario pemindaian nomor pesanan yang valid, sudah diproses, salah, dan tanpa ID picker menggunakan metode EP. TC31 adalah skenario positif jika nomor pesanan terdaftar di "Packing - Belum Mulai", sistem diharapkan menerima input dan menampilkan detail pesanan. TC32 hingga TC34 adalah skenario negatif jika nomor pesanan sudah diproses, salah, atau tanpa ID picker, diharapkan sistem menolak input dan menampilkan pesan kesalahan.

TC35 hingga TC40 menguji skenario input qty pack menggunakan metode BVA dan EP. TC35 hingga TC37 adalah skenario negatif dan positif untuk input qty pack yang kosong, sesuai, atau lebih besar dari qty pesanan, diharapkan sistem menerima atau menolak input sesuai batasan yang ditentukan. TC38 hingga TC40 menguji skenario input qty pack yang lebih kecil, kosong, atau dengan spasi tambahan, diharapkan sistem menolak input dan menampilkan pesan kesalahan.

Tabel 3. 6 Skenario Pengujian - Buat Pengiriman Baru

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC41	Klik button Buat Pengiriman dengan memilih pesanan yang memiliki kurir	EP	Valid	Berhasil masuk ke halaman Buat Pengiriman
TC42	Klik button Buat Pengiriman dengan memilih pesanan yang tidak memiliki kurir	EP	Invalid	Gagal masuk ke halaman Buat Pengiriman dan menampilkan error message " <i>Pesanan tidak memiliki kurir</i> "

TC43	Klik button Buat Pengiriman dengan memilih pesanan yang tidak memiliki dan memiliki kurir	EP	Invalid	Gagal masuk ke halaman Buat Pengiriman dan menampilkan error message <i>“Lokasi dan kurir dari pesanan terpilih harus sama”</i>
TC44	Klik button Buat Pengiriman dengan pesanan yang memiliki kurir/Lokasi berbeda	EP	Invalid	Gagal masuk ke halaman Buat Pengiriman dan menampilkan error message <i>“Lokasi dan kurir dari pesanan terpilih harus sama”</i>
TC45	Klik button Buat Pengiriman dengan tidak memilih pesanan	EP	Invalid	Button Buat Pengiriman tidak muncul
TC46	Scan id shipper dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama shipper
TC47	Scan id shipper dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC48	Scan id shipper dengan mengosongkan input	BVA	Min-1	Sistem menolak inputan dan menampilkan error message <i>“Tidak boleh kosong”</i>
TC49	Scan id shipper dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC50	Scan id shipper dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC51	Buat pengiriman dengan mengubah nomor pengiriman	EP	Valid	Berhasil membuat kartu pengiriman baru yang berisi pesanan dipilih
TC52	Buat pengiriman dengan jumlah pesanan 1 atau lebih	BVA	Min	Berhasil membuat kartu pengiriman baru yang berisi pesanan dipilih

TC53	Buat pengiriman dengan jumlah pesanan = 0	BVA	Min-1	Gagal membuat pengiriman dan menampilkan error message " <i>Minimal 1 pesanan</i> "
TC54	Buat pengiriman dengan input nomor pengiriman berjumlah 1 karakter	BVA	Min	Berhasil membuat kartu pengiriman baru yang berisi pesanan dipilih
TC55	Buat pengiriman dengan mengosongkan nomor pengiriman	BVA	Min-1	Gagal membuat pengiriman dan menampilkan error message " <i>Tidak boleh kosong</i> "
TC56	Buat pengiriman dengan input nomor pengiriman berjumlah 50 karakter	BVA	Max	Berhasil membuat kartu pengiriman baru yang berisi pesanan dipilih
TC57	Buat pengiriman dengan input no pengiriman berjumlah lebih dari 50 karakter	BVA	Max+1	Gagal membuat pengiriman dan menampilkan error message " <i>Maksimal nomor pengiriman 50 karakter</i> "
TC58	Memilih tanggal pengiriman sesuai tanggal pembuatan atau sesudah	EP	Valid	Sistem menerima inputan
TC59	Memilih tanggal pengiriman sebelum tanggal pembuatan	EP	Invalid	Sistem disable tanggal sebelum tanggal pembuatan

Pada tabel 3.6 merupakan tabel skenario pengujian pada fitur Buat Pengiriman di tab Shipping – Siap Kirim. Terdapat 19 skenario pengujian, diantaranya 12 skenario menggunakan metode EP dan tujuh skenario menggunakan metode BVA. Skenario awal pada TC41 memverifikasi jika tombol Buat Pengiriman ditekan setelah memilih pesanan yang memiliki kurir, dan hasil yang diharapkan adalah berhasil masuk ke halaman Buat Pengiriman. Sebaliknya, TC42 hingga TC45 mencakup berbagai skenario negatif seperti memilih pesanan tanpa kurir, dengan kurir dan

lokasi yang tidak sesuai, atau tanpa memilih pesanan sama sekali. Pada skenario ini, sistem diharapkan menolak input dan memberikan pesan kesalahan yang sesuai.

Kemudian, TC46 dan TC47 menguji pemindaian ID shipper dengan input yang valid dan tidak valid. TC46 memeriksa ID shipper yang terdaftar, dengan hasil yang diharapkan adalah sistem menerima input dan menampilkan nama shipper. Sementara itu, TC47 berfokus pada skenario negatif jika ID shipper tidak terdaftar, yang seharusnya menghasilkan pesan kesalahan "Data tidak ditemukan".

TC48 hingga TC51 menguji skenario pemindaian ID shipper dengan input kosong, atau dengan spasi tambahan di depan dan di belakang. Skenario ini, diharapkan sistem menolak input dan menampilkan pesan kesalahan yang sesuai.

Sementara itu, TC52 hingga TC55 mengevaluasi pembuatan pengiriman dengan jumlah pesanan yang valid, lebih dari nol, dan nol. TC52 dan TC53 menguji skenario positif jika jumlah pesanan valid, diharapkan sistem berhasil membuat kartu pengiriman. TC54 menguji skenario negatif dengan jumlah pesanan nol, yang diharapkan sistem menolak dengan pesan kesalahan "Minimal 1 pesanan".

Selanjutnya pada TC56 dan TC57 menggunakan metode BVA untuk menguji panjang maksimum nomor pengiriman. TC56 menunjukkan hasil positif dengan nomor pengiriman berjumlah 50 karakter, sementara TC57 menyoroti skenario negatif dengan nomor pengiriman lebih dari 50 karakter, maka sistem menolak input dengan pesan kesalahan "Maksimal nomor pengiriman 50 karakter".

Terakhir, TC58 dan TC59 menguji pemilihan tanggal pengiriman. TC58 mencakup skenario positif jika tanggal pengiriman sesuai atau setelah tanggal pembuatan, diharapkan sistem menerima input. TC59 menyoroti skenario negatif dengan tanggal pengiriman sebelum tanggal pembuatan, jika sistem menolak input dan memberikan pesan kesalahan.

Tabel 3. 7 Skenario Pengujian - Buat Pengiriman Terpilih

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
----	--------------------	--------	-------	-----------------------

TC60	Scan id shipper dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama shipper
TC61	Scan id shipper dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC62	Scan id shipper dengan mengosongkan input	BVA	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Tidak boleh kosong</i> "
TC63	Scan id shipper dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC64	Scan id shipper dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC65	Pilih nomor pengiriman	EP	Invalid	Sistem menerima inputan
TC66	Mengosongkan nomor pengiriman	EP	Invalid	Sistem menampilkan error message " <i>Tidak boleh kosong</i> "
TC67	Simpan pengiriman dengan jumlah pesanan = 1 atau lebih	BVA	Min	Berhasil memasukkan pesanan ke kartu pengiriman
TC68	Simpan pengiriman dengan jumlah pesanan = 0	BVA	Min-1	Gagal memasukkan pesanan dan menampilkan error message " <i>Minimal 1 pesanan</i> "

Pada tabel 3.7 merupakan tabel skenario pengujian pada fitur Buat Pengiriman Terpilih pada tab Shipping – Siap Kirim. Terdapat sembilan skenario pengujian dengan enam skenario menggunakan metode EP dan tiga skenario menggunakan metode BVA. Diawali dengan TC60, skenario ini memverifikasi scan ID shipper dengan input ID yang valid, sistem diharapkan menerima input dan menampilkan nama shipper. Sementara itu, TC61 berfokus pada skenario negatif jika ID shipper

tidak terdaftar, dengan hasil yang diharapkan adalah sistem menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

Selanjutnya, TC62 hingga TC64 menguji skenario scan ID shipper dengan input kosong atau dengan spasi tambahan di depan dan di belakang. Dalam skenario ini, sistem diharapkan menolak input dan menampilkan pesan kesalahan yang sesuai.

Lalu pada skenario pemilihan nomor pengiriman, TC65 menguji skenario positif jika nomor pengiriman dipilih, dengan hasil yang diharapkan adalah sistem menerima input. Sebaliknya, TC66 mengevaluasi skenario negatif jika nomor pengiriman dikosongkan, maka sistem diharapkan menolak input dan menampilkan pesan kesalahan "Tidak boleh kosong".

Kemudian, TC67 dan TC68 fokus pada skenario penyimpanan pengiriman dengan jumlah pesanan menggunakan metode BVA. TC67 menguji skenario positif dengan jumlah pesanan satu atau lebih, maka sistem diharapkan berhasil memasukkan pesanan ke kartu pengiriman. TC68, sebaliknya, menguji skenario negatif dengan jumlah pesanan nol, maka sistem diharapkan gagal memasukkan pesanan dan menampilkan pesan kesalahan "Minimal 1 pesanan".

Tabel 3. 8 Skenario Pengujian – Tambah Kartu Pengiriman Baru

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC69	Buat pengiriman dengan input nomor pengiriman berjumlah 1 karakter	BVA	Min	Berhasil membuat kartu pengiriman baru
TC70	Buat pengiriman dengan mengosongkan nomor pengiriman	BVA	Min-1	Gagal membuat pengiriman dan menampilkan error message " <i>Tidak boleh kosong</i> "
TC71	Buat pengiriman dengan input nomor pengiriman berjumlah 50 karakter	BVA	Max	Berhasil membuat kartu pengiriman baru
TC72	Buat pengiriman dengan input no	BVA	Max+1	Gagal membuat pengiriman dan

	pengiriman berjumlah lebih dari 50 karakter			menampilkan error message " <i>Maksimal nomor pengiriman 50 karakter</i> "
TC73	Pilih kurir	EP	Valid	Sistem menerima inputan
TC74	Mengosongkan kurir	EP	Invalid	Sistem menampilkan error message " <i>Tidak boleh kosong</i> "
TC75	Pilih lokasi	EP	Valid	Sistem menerima inputan
TC76	Mengosongkan kurir	EP	Invalid	Sistem menampilkan error message " <i>Tidak boleh kosong</i> "
TC77	Memilih tanggal pengiriman sesuai tanggal pembuatan atau sesudah	EP	Valid	Sistem menerima inputan
TC78	Memilih tanggal pengiriman sebelum tanggal pembuatan	EP	Invalid	Sistem disable tanggal sebelum tanggal pembuatan

Pada tabel 3.8 merupakan tabel skenario pengujian untuk fitur Tambah Pengiriman Baru atau buat kartu pengiriman baru di tab Shipping – Siap Kirim. Terdapat 10 skenario pengujian dengan enam skenario menggunakan metode EP dan empat skenario menggunakan metode BVA. Dimulai dengan TC69, skenario ini memverifikasi pembuatan pengiriman dengan input nomor pengiriman berjumlah satu karakter. Dalam skenario ini, hasil yang diharapkan adalah sistem berhasil membuat kartu pengiriman baru. Berlanjut ke TC70, skenario ini mengevaluasi input nomor pengiriman yang dikosongkan, maka sistem diharapkan menolak input dan menampilkan pesan kesalahan "Tidak boleh kosong".

TC71 menguji pembuatan pengiriman dengan input nomor pengiriman berjumlah 50 karakter, dan hasil yang diharapkan adalah sistem berhasil membuat kartu pengiriman baru. Di sisi lain, TC72 berfokus pada skenario negatif jika nomor pengiriman melebihi 50 karakter, dan sistem diharapkan menolak input dengan pesan kesalahan "Maksimal nomor pengiriman 50 karakter".

Beralih ke skenario pemilihan kurir, TC73 memeriksa skenario positif jika kurir dipilih, sehingga sistem diharapkan menerima input. Sebaliknya, TC74

mengevaluasi kondisi jika kurir dikosongkan, diharapkan sistem menolak input dengan pesan kesalahan "Tidak boleh kosong".

Selanjutnya, TC75 dan TC76 fokus pada pemilihan lokasi. TC75 memastikan bahwa pemilihan lokasi yang valid diterima oleh sistem, sedangkan TC76 menguji kondisi negatif jika lokasi dikosongkan, dan sistem diharapkan menampilkan pesan kesalahan yang sesuai.

Terakhir yaitu pengujian tanggal pengiriman, TC77 dan TC78 menguji validitas tanggal pengiriman. TC77 memverifikasi bahwa pemilihan tanggal pengiriman yang sesuai dengan tanggal pembuatan atau setelahnya diterima oleh sistem. Sebaliknya, TC78 mengevaluasi skenario negatif jika tanggal pengiriman dipilih sebelum tanggal pembuatan, jika sistem diharapkan menolak input dan menampilkan pesan kesalahan.

Tabel 3. 9 Skenario Pengujian - Masukkan Pengiriman

ID	Skenario Pengujian	Metode	Nilai	Hasil yang diharapkan
TC79	Scan id shipper dengan input id yang terdaftar	EP	Valid	Sistem menerima inputan dan menampilkan nama shipper
TC80	Scan id shipper dengan input id yang tidak terdaftar	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC81	Scan id shipper dengan mengosongkan input	BVA	Min-1	Sistem menolak inputan dan menampilkan error message " <i>Tidak boleh kosong</i> "
TC82	Scan id shipper dengan menambahkan spasi di belakang	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "
TC83	Scan id packer dengan menambahkan spasi di depan	EP	Invalid	Sistem menolak inputan dan menampilkan error message " <i>Data tidak ditemukan</i> "

TC84	Pilih nomor pengiriman tanpa scan id shipper	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Silahkan isi shipper terlebih dahulu”</i>
TC85	Scan nomor pesanan tanpa scan id shipper	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Silahkan isi shipper terlebih dahulu”</i>
TC86	Scan nomor pesanan dengan memilih nomor pengiriman	EP	Valid	Berhasil memasukkan pesanan ke pengiriman yang dipilih user
TC87	Scan nomor pesanan tanpa memilih nomor pengiriman	EP	Valid	Berhasil memasukkan pesanan ke pengiriman yang dipilih oleh sistem secara random.
TC88	Scan nomor pesanan dengan input benar	EP	Valid	Berhasil memasukkan pesanan ke pengiriman
TC89	Scan nomor pesanan dengan input salah	EP	Invalid	Sistem menolak inputan dan menampilkan error message <i>“Data tidak ditemukan”</i>
TC90	Scan nomor pesanan yang belum siap kirim	EP	Invalid	Gagal memasukkan pesanan dan menampilkan error message <i>“Pesanan belum siap kirim”</i>
TC91	Scan nomor pesanan dengan pesanan yang sudah dikirim atau selesai	EP	Invalid	Gagal memasukkan pesanan dan menampilkan error message <i>“Pesanan ini sudah dikirim”</i>
TC92	Scan nomor pesanan yang tidak memiliki kurir	EP	Invalid	Gagal memasukkan pesanan dan menampilkan error message <i>“Pesanan ini tidak memiliki kurir”</i>
TC93	Scan nomor pesanan dengan lokasi/kurir yang tidak sesuai pengiriman dipilih	EP	Invalid	Gagal memasukkan pesanan dan menampilkan error message <i>“Kurir atau lokasi pesanan tidak sesuai dengan jadwal pengiriman yang dipilih”</i>

TC94	Scan nomor pesanan yang sudah masuk kartu pengiriman	EP	Invalid	Gagal memasukkan pesanan dan menampilkan error message <i>“Pesanan sudah terjadwal”</i>
TC95	Scan nomor pesanan dengan mengosongkan input	BVA	Min-1	Sistem menolak inputan dan menampilkan error message <i>“Tidak boleh kosong”</i>

Pada tabel 3.9 merupakan tabel skenario pengujian pada fitur Masukkan Pengiriman di tab Shipping – Siap Kirim. Berdasarkan tabel, terdapat 15 skenario menggunakan metode EP dan dua skenario menggunakan metode BVA. Dimulai dengan TC79, skenario ini memverifikasi pemindaian ID shipper dengan input ID yang valid, maka sistem diharapkan menerima input dan menampilkan nama shipper. Sementara itu, TC80 berfokus pada skenario negatif jika ID shipper tidak terdaftar, dengan hasil yang diharapkan adalah sistem menolak input dan menampilkan pesan kesalahan "Data tidak ditemukan".

Berlanjut ke TC81, skenario ini menguji pemindaian ID shipper dengan input yang dikosongkan, maka sistem diharapkan menolak input dan menampilkan pesan kesalahan "Tidak boleh kosong". Sebaliknya, TC82 hingga TC84 mengevaluasi pemindaian ID shipper atau ID packer dengan spasi tambahan di depan atau di belakang, serta memilih nomor pengiriman tanpa ID shipper. Pada skenario ini, sistem diharapkan menolak input dan memberikan pesan kesalahan yang sesuai.

TC85 dan TC86 menguji pemindaian nomor pesanan dengan dan tanpa pemilihan nomor pengiriman. Dalam skenario TC86, sistem diharapkan berhasil memasukkan pesanan ke pengiriman yang dipilih user, sementara TC85 menyoroti skenario negatif jika nomor pesanan dipindai tanpa ID shipper, yang diharapkan sistem menolak input dan menampilkan pesan kesalahan "Silakan isi shipper terlebih dahulu".

Selanjutnya, TC87 hingga TC90 memverifikasi berbagai kondisi pemindaian nomor pesanan, seperti input benar, input salah, pesanan yang belum siap kirim, dan pesanan yang sudah dikirim atau selesai. Sistem diharapkan menerima input

yang valid dan menolak input yang tidak valid, serta memberikan pesan kesalahan yang sesuai.

TC91 hingga TC95 fokus pada kondisi pesanan yang tidak memiliki kurir, lokasi yang tidak sesuai, atau pesanan yang sudah masuk kartu pengiriman. Dalam skenario ini, sistem diharapkan menolak input dan memberikan pesan kesalahan yang tepat.

Pada pengujian aplikasi omnichannel ini terdapat 8 lingkup fitur yang diuji, yaitu :

- Fitur 1 : Login User
- Fitur 2 : Cetak Picklist
- Fitur 3 : Serahkan ke Picker
- Fitur 4 : Mulai Packing
- Fitur 5 : Buat Pengiriman Baru
- Fitur 6 : Buat Pengiriman Terpilih
- Fitur 7 : Tambah Pengiriman Baru
- Fitur 8 : Masukkan Pengiriman

5. Environment Setup

Tahap ini dilakukan untuk menyiapkan lingkungan pengujian dan dikonfigurasi sesuai kebutuhan. Ini termasuk pengaturan perangkat keras, perangkat lunak, jaringan, data pengujian, dan sistem lain yang diperlukan untuk menjalankan kasus pengujian. Persiapan ini dilakukan secara paralel dengan tahap analisis dan desain.

6. Test Execution

Tahap ini merupakan tahap utama dalam penelitian yaitu mengeksekusi pengujian dengan menjalankan pengujian berdasarkan *test case* yang telah disusun dan menerapkan metode *equivalence partitioning* serta *boundary value analysis* pada eksekusi pengujian tersebut. Selain menjalankan kasus pengujian, peneliti akan mencatat hasil pengujian termasuk keberhasilan, kegagalan, dan anomali yang ditemukan.

7. *Test Cycle Closure*

Pada tahap terakhir ini, peneliti akan melakukan analisis terhadap hasil pengujian untuk menilai kesesuaian sistem dengan spesifikasi dan mengidentifikasi area yang memerlukan perbaikan. Akan dilakukan juga analisis tingkat efektivitas berdasarkan hasil pengujian *Omnichannel Jubelio*. Semakin besar nilai efektivitas tersebut, maka semakin baik fungsionalitas dan kegunaan sistem. Nilai efektivitas adalah hasil persentase kelayakan aplikasi. Nilai efektivitas dapat dihitung dengan rumus efektivitas tiap tabel fitur dengan rumus :

$$\text{Nilai Efektivitas Fitur} = \left(\frac{\sum \text{Pengujian Berhasil}}{\sum \text{Skenario Pengujian}} \right) \times 100\%$$

Pada rumus di atas merupakan rumus untuk menghitung nilai efektivitas dari setiap tabel skenario. Nilai efektivitas per tabel ini berguna untuk mengetahui tingkat fungsionalitas dari masing-masing tabel skenario yang diuji, Setelah itu, akan dilakukan perhitungan nilai efektivitas keseluruhan aplikasi dengan rumus sebagai berikut :

$$\text{Nilai Efektivitas Aplikasi} = \left(\frac{\sum \text{Nilai efektivitas semua fitur}}{\sum \text{Jumlah fitur}} \right) \times 100\%$$

Hasil dari perhitungan nilai efektivitas keseluruhan aplikasi dapat menentukan tingkat keberhasilan dari aplikasi *omnichannel Jubelio*. Efektivitas dapat diukur menggunakan standar sesuai dengan acuan Litbang Depdagri seperti pada tabel di bawah :

Rasio Efektivitas	Tingkat Capaian
Dibawah 40	Sangat tidak efektif
40 - 59,99	Tidak efektif
60 – 79,99	Cukup efektif
Di atas 80	Sangat efektif

Sumber : Litbang Depdagri, 1991 dalam Budiani 2009

Setelah itu, dilakukan penyusunan laporan pengujian secara menyeluruh, termasuk deskripsi proses pengujian, temuan, analisis, dan rekomendasi untuk perbaikan

lebih lanjut. Rekomendasi perbaikan sangat dibutuhkan bagi aplikasi yang masih memiliki kesalahan atau bug agar segera diperbaiki. Pemberian rekomendasi harus dilakukan dengan kehati-hatian dan tepat dengan menentukan tingkat keparahan dan skala prioritas yang akurat dan jelas untuk membantu kinerja perusahaan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengujian aplikasi omnichannel Jubelio dengan menggunakan metode pengujian Boundary Value Analysis dan Equivalence Partitioning berhasil dilakukan. Berdasarkan hasil pengujian pada aplikasi, maka dapat disimpulkan sebagai berikut :

1. Pengujian black box pada aplikasi omnichannel Jubelio menerapkan metode Equivalence Partitioning lebih banyak dibandingkan metode Bounday Value Analysis dikarenakan banyak fungsi aplikasi yang tidak memiliki batasan numerik, sehingga EP lebih efektif untuk mengidentifikasi masalah fungsional secara umum. Teknik EP digunakan untuk menguji validitas input dengan kriteria tertentu dengan mengelompokkan nilai input ke dalam kelas valid dan tidak valid, sedangkan teknik BVA digunakan untuk menguji input pada batasan tertentu yaitu batas atas dan bawah nilai input.
2. Hasil pengujian aplikasi dilakukan pada delapan fitur aplikasi atau tabel skenario pengujian dengan total 95 skenario pengujian di dalamnya. Jumlah skenario yang berhasil yaitu 82 skenario, sedangkan yang gagal berjumlah 13 skenario. Maka, nilai efektivitas aplikasi yang didapat dari pengujian tersebut sebesar 86,31% yang artinya memiliki nilai efektivitas yang cukup (lebih dari 70%), tetapi perlu ditingkatkan lagi terhadap beberapa kekurangan yang ada.
3. Berdasarkan temuan pengujian, terdapat lima rekomendasi perbaikan yang dapat diberikan kepada tim pengembang. Rekomendasi ini meliputi perbaikan validasi input pada form yang masih dapat dikosongkan, penambahan pesan kesalahan yang lebih informatif, dan peningkatan validasi jenis data input. Implementasi perbaikan ini diharapkan dapat meningkatkan kualitas aplikasi Omnichannel Jubelio, meningkatkan kepuasan pelanggan, dan efisiensi operasional.

5.2 Saran

Penelitian yang dilakukan masih banyak memiliki kekurangan, sehingga diperlukan beberapa perbaikan untuk memperoleh hasil yang lebih optimal, seperti :

1. Pengujian dapat dilakukan menggunakan teknik black-box lainnya untuk mengetahui dan mengevaluasi efektivitas teknik tersebut dengan teknik lainnya.
2. Pengujian dapat menggunakan tools automasi seperti Selenium IDE atau Katalon agar pengujian lebih cepat dan praktis.

DAFTAR PUSTAKA

- [1] M. Komarudin MZ, “Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah,” *Jurnal Manajemen Informatika*, vol. 6, no. 1, 2016, [Daring]. Tersedia pada: <https://ojs.ummetro.ac.id/index.php/mikrotik/article/view/303>
- [2] M. Last, S. Eyal, dan A. Kandel, “Effective black-box testing with genetic algorithms,” *Lecture Notes in Computer Science*, vol. 3875, hlm. 134–148, 2006, doi: 10.1007/11678779_10.
- [3] T. Snadhika Jaya, “Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung),” *Jurnal Informatika Politeknik Harapan Bersama*, vol. 3, no. 1, hlm. 45–48, 2018, doi: 10.30591/JPIT.V3I1.647.
- [4] M. Sholeh, I. Gisfas, Cahiman, dan M. A. Fauzi, “Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods,” *J Phys Conf Ser*, vol. 1823, no. 1, 2021, doi: 10.1088/1742-6596/1823/1/012029.
- [5] G. Suharyono Farra Wardah, Kartini, dan A. Junaidi, “Penerapan Metode Boundary Value Analysis dan Equivalence Partitioning dalam Pengujian Black Box untuk Aplikasi Siadita,” *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 1, 2024.
- [6] A. E. Rahmawati, H. E. Wahanani, dan F. P. Aditiawan, “Pengujian Aplikasi E Buk Kades Menggunakan Teknik Boundary Value Analysis dan Equivalence Partitioning,” *Jurnal Informatika*, vol. 12, no. 02, 2023.

- [7] P. C. Verhoef, P. K. Kannan, dan J. J. Inman, “From Multi-Channel Retailing to Omni-Channel Retailing: Introduction to the Special Issue on Multi-Channel Retailing,” *Journal of Retailing*, vol. 91, no. 2, hlm. 174–181, Jun 2015, doi: 10.1016/J.JRETAI.2015.02.005.
- [8] C. Lazaris, A. Vrechopoulos, K. Fraidaki, dan G. Doukidis, “Exploring the ‘Omnichannel’ Shopper Behaviour,” *AMA SERV SIG, International Service Research Conference*, 2014, doi: 10.13140/2.1.1278.2089.
- [9] D. K. Rigby, “The Future of Shopping,” *Harvard Business Review*. Diakses: 14 November 2023. [Daring]. Tersedia pada: <https://hbr.org/2011/12/the-future-of-shopping>
- [10] I. C. S. S. C. Committee, “IEEE Standard Glossary of Software Engineering Terminology,” *IEEE Std 610.12-1990*, hlm. 1–84, 1990, doi: 10.1109/IEEESTD.1990.101064.
- [11] A. Arfan, “Software Quality Assurance Dengan Metode Pengujian Black Box (Studi Kasus : PT Git Solution),” 2022.
- [12] J. Supardi dan R. Ariani Sukanto, “Black-Box Testing,” 2017.
- [13] I. R. Dhaifullah, M. M. H, A. A. Salsabila, dan M. A. Yaqin, “Survei Teknik Pengujian Software,” *Journal Automation Computer Information System*, vol. 2, no. 1, hlm. 31–38, 2022, doi: 10.47134/JACIS.V2I1.42.
- [14] A. N. Hasibuan dan T. Dirgahayu, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna,” *AUTOMATA*, vol. 2, no. 1, 2021, [Daring]. Tersedia pada: <https://journal.uui.ac.id/AUTOMATA/article/view/17367>
- [15] T. Hidayat dan M. Muttaqin, “Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis,” *Jurnal Teknik Informatika UNIS JUTIS*, vol. 6, no. 1, hlm. 2252–5351, 2018.

- [16] geeksforgeeks, “Equivalence Partitioning Method.” Diakses: 1 Agustus 2024. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/equivalence-partitioning-method/>
- [17] M. Albarka Umar dan C. Zhanfang, “A Comparative Study of Dynamic Software Testing Techniques,” *Int. J. Advanced Networking and Applications*, vol. 12, no. 03, hlm. 4575–4584, 2020.
- [18] U. Hanifah, R. Alit, dan Sugiarto, “Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk,” *Jurnal Teknologi Informasi dan Komunikasi*, vol. XI, 2016.
- [19] M. S. Mustaqbal, R. F. Firdaus, dan H. Rahmadi, “Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 1, no. 3, 2015, doi: 10.33197/JITTER.VOL1.ISS3.2015.62.
- [20] Y. F. Kurnia dan D. T. Yulianti, “Analisis Proses Software Testing pada PT. Signify Commercial Indonesia,” *Jurnal STRATEGI*, vol. 3, no. 1, hlm. 178–186, 2021, [Daring]. Tersedia pada: <https://strategi.it.maranatha.edu/index.php/strategi/article/view/245>
- [21] D. A. Syuhada dan A. Nursikuwagus, “Pengujian Aplikasi Perhitungan Stok Barang Dengan Metode Blackbox Pada CV. Delta Pilar (Cabang Bandung),” 2019.
- [22] P. Ayu Desi Anggara Santi, R. Afwani, dan M. A. Albar, “Pengujian Black Box Dengan Metode Equivalence Partitioning dan Boundary Value Analysis (Studi Kasus : Sistem Informasi Akademik Universitas Mataram),” 2022.
- [23] F. I. Pratama, E. M. N. Subroto, R. M. Haira, dan M. A. Yaqin, “Pengujian Black Box pada Aplikasi E-Commerce OpenCart dengan Metode Equivalence Partitioning dan Boundary Value Analysis,” *Jurnal Ilmiah Informatika*, vol. 8, no. 1, hlm. 54–64, 2023, doi: 10.35316/JIMI.V8I1.54-64.

- [24] S. Adam, “Equivalence Partitions pada BlackBox Testing terhadap Sistem Pemberdayaan Pemerataan UMKM,” *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 2, hlm. 382–387, 2021, doi: 10.32493/informatika.v6i2.11457.
- [25] M. Ilham, D. Apriansyah, M. R. Nurhakiki, N. Apriani, dan A. Saifudin, “Pengujian Black Box pada Aplikasi Ecommerce Berbasis Website Menggunakan Teknik Equivalence Partitions,” *OKTAL: Jurnal Ilmu Komputer dan Science Pengujian*, vol. 2, no. 7, hlm. 1958–1966, 2023, [Daring]. Tersedia pada: <https://journal.mediapublikasi.id/index.php/oktal>