

# Black Box Testing Menggunakan Boundary Value Analysis dan Equivalence Partitioning pada Aplikasi Pengadaan Bahan Baku Batik dengan Pendekatan Use Case

I Dewa Made Widia<sup>1</sup>, Sovia Rosalin<sup>2</sup>, Salnan Ratih Asriningtias<sup>3</sup>, Elta Sonalita<sup>4</sup>

<sup>1,2,3</sup>Program Pendidikan Vokasi – Universitas Brawijaya

<sup>4</sup>Teknik Elektro – Universitas Merdeka Malang

<sup>1</sup>dewa\_vokasi@ub.ac.id

<sup>2</sup>soviavokasi@ub.ac.id

<sup>3</sup>salnanratih@gmail.com

<sup>4</sup>elta.sonalitha@unmer.ac.id

Received : 23-02-2021; Accepted: 13-03-2022; Published: 13-03-2022

**Abstrak**— Aplikasi pengadaan bahan baku batik merupakan aplikasi website yang akan digunakan oleh UMKM Batik untuk dapat membantu manager pembelian mengambil keputusan dalam menentukan jumlah order bahan baku. Untuk memastikan aplikasi sesuai kebutuhan fungsional yang diharapkan maka diperlukan pengujian. Pengujian dilakukan menggunakan Black Box Testing yaitu melakukan validasi output dari data input yang diberikan. Test case pada black box testing didesain menggunakan pendekatan use case, karena kebutuhan fungsional aplikasi digambarkan dalam use case diagram. Desain test case yang dapat membantu dalam menemukan kesalahan aplikasi merupakan pertimbangan penting dalam pengujian aplikasi. Terdapat beberapa jenis metode yang dapat digunakan dalam menentukan test case diantaranya Boundary Value Analysis dan Equivalence Partitioning. Boundary value analysis hanya dapat digunakan untuk menguji tipe data dengan nilai rentang. Sedangkan Equivalence partitioning digunakan untuk mencari semua kemungkinan data berdasarkan kriteria tertentu. Maka dalam penelitian ini pengujian dilakukan dengan menggabungkan Boundary Value Analysis dan Equivalence Partitioning. Hasil pengujian menunjukkan metode dapat menemukan kesalahan dari aplikasi secara efektif hal ini dibuktikan dengan nilai DRE yang diperoleh sebesar 0.45 yang berarti 45% test case yang dibangun tidak lolos uji.

**Kata kunci**— Black Box Testing, Boundary Analysis Value, Equivalence Partitioning, UMKM, Use Case

**Abstract**— The batik raw material purchase order application is a website application that will be used by Batik MSMEs to be able to help managers make decisions in determining the quantity of raw material orders. To ensure the application meets the expected functional requirements, testing is required. Testing is done using Black Box Testing, which is validating the output from the given data input. Test cases in black box testing can be designed using use cases, because the functional requirements of the application are described in the use case diagram. Test case design that can assist in finding application errors are an important consideration in application testing. There are several types of methods that can be used in determining test cases including Boundary Value Analysis and Equivalence Partitioning. Boundary value analysis can only be used to test data types with range

values. Whereas the Equivalence partition is used to exploit all possible data based on defined criteria. So in this study the test was carried out by combining Boundary Value Analysis and Equivalence Partitioning. The test results show the method can find errors from effective applications, this is evidenced by the DRE value obtained of 0.45, which means that 45% of the test cases built did not pass the test.

**Keyword**— Black Box Testing, Boundary Analysis Value, Equivalence Partitioning, MSMEs, Use Case

## I. PENDAHULUAN

Pengujian merupakan bagian yang tidak terpisahkan dari pengembangan aplikasi. Pengujian merupakan serangkaian aktivitas untuk menemukan kesalahan dalam isi, fungsi, kegunaan, kemampuan navigasi, kinerja, dan keamanan aplikasi [1]. Pentingnya pengujian aplikasi dan implikasinya mengacu pada penjaminan mutu dan kualitas aplikasi. Menurut ISO-9216, salah satu karakteristik aplikasi dikatakan berkualitas adalah aplikasi mampu bekerja secara fungsional yaitu pengujian yang memastikan aplikasi yang dibuat dapat berjalan sesuai dengan kebutuhan fungsional yang diharapkan [2].

Salah satu jenis pengujian yang berfokus pada kebutuhan fungsional dari aplikasi adalah black box testing. Manfaat dari black box testing dapat mengetahui seberapa baik dan kekurangan sistem melaksanakan fungsinya serta interaksi user terhadap aplikasi. Pada black box testing diperlukan perancangan test case yang berfungsi untuk mengidentifikasi dan memberikan kondisi – kondisi yang akan di implementasikan dalam pengujian, untuk meyakinkan bahwa aplikasi dapat memenuhi kebutuhan fungsional user [3].

Kebutuhan fungsional aplikasi dapat digambarkan dalam bentuk use case. Use case merupakan skenario yang menggambarkan bagaimana interaksi user kepada aplikasi. Use case memberikan informasi mengenai apa yang diharapkan oleh user dan mengenai apa yang akan dibangun oleh pengembang, maka use case bisa dijadikan bahan pertimbangan dalam perancangan test case [4]. Beberapa peneliti mencoba mendesain test case dengan

pendekatan use case dan memberikan hasil yang signifikan [5][6].

Pertimbangan terpenting dalam pengujian aplikasi adalah mendesain test case yang efektif. Test case dikatakan efektif jika memiliki kemampuan dalam menemukan kesalahan aplikasi. Strategi yang baik adalah mendesain test case yang lengkap, tetapi hal ini akan memakan waktu pengujian yang sangat lama [7]. Untuk itu perlu metode untuk mengurangi jumlah test case dengan tidak menghilangkan test case yang berpotensi menemukan kesalahan. Terdapat beberapa jenis metode yang dapat digunakan dalam menentukan test case dalam black box testing diantaranya Boundary Value Analysis dan Equivalence Class Partitioning [8].

Peneliti terdahulu banyak yang menggunakan kedua metode tersebut, baik Boundary Value Analysis [9][10] maupun Equivalence Partitioning [11][12]. Dari kedua metode tersebut, masing-masing memiliki kelemahan dan kelebihan [13]. Berdasarkan hasil perbandingan pada penelitian [14] metode equivalence partitioning untuk menguji data dengan tipe data nilai rentang dan untuk mencari semua kemungkinan data berdasarkan kriteria tertentu. Sedangkan boundary value analysis hanya dapat untuk menguji tipe data dengan nilai rentang dan memiliki nilai yang lebih jelas.

Aplikasi pengadaan bahan baku batik merupakan aplikasi berbasis web yang akan digunakan oleh pelaku UMKM Batik untuk dapat membantu dalam proses order atau pembelian bahan baku ke supplier. Selain menyimpan dan membantu mempermudah dalam mengolah data dan menampilkan laporan pembelian bahan baku, aplikasi ini dilengkapi fitur untuk membantu manager pembelian mengambil keputusan dalam menentukan jumlah order bahan baku yang akan dibeli serta memberikan rekomendasi pemilihan supplier untuk bahan baku yang akan dibeli menggunakan FAHP dan Fuzzy Tsukamoto. Modul dalam aplikasi ini diantaranya modul data bahan baku, modul data supplier, modul data persediaan bahan baku dan modul purchase order.

Aplikasi ini masih dalam tahap pengujian. Untuk memastikan aplikasi sudah memenuhi kebutuhan dari UMKM dan siap untuk digunakan maka diperlukan pengujian sesuai kebutuhan fungsional yang sudah ditentukan sebelumnya atau sesuai diharapkan. Pengujian dilakukan menggunakan Black Box Testing yaitu melakukan validasi output apakah sesuai dengan yang diharapkan dari data input yang diberikan. Supaya jumlah data test case tidak terlalu banyak, maka jumlah test case perlu dibatasi dengan tidak menghilangkan test case yang memiliki probability menemukan kesalahan. Test case di desain dari skenario use case diagram yang sudah dibuat. Akan tetapi tipe data input test case tidak hanya data dengan rentang saja tetapi juga terdapat kriteria tertentu yang tidak dalam bentuk rentang. Maka dalam penelitian ini pengujian dilakukan dengan menggabungkan Boundary Value Analysis untuk menguji data input dengan rentang dengan Equivalence Partitioning untuk menguji data input dengan kriteria tertentu.

## II. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian terapan untuk menguji aplikasi pengadaan bahan baku pada UMKM batik. Beberapa tahap pengujian yang dilaksanakan adalah sebagai berikut [15]:

### 1. Desain Use Case Diagram

Tahap ini meliputi mengidentifikasi *user*, mengidentifikasi *use case*, dan mendesain *use case diagram*.

### 2. Desain Test Case

Tahap ini meliputi membuat skenario dari masing-masing *use case*, membuat *test case matrix*, desain nilai *test case* menggunakan *boundary value analysis* dan *equivalence partitioning*. Batas pada pengelompokan data test case pada *boundary value analysis* adalah sebagai berikut :

- BLB (*Below the Lower Bound*) atau nilai dibawah batas minimum.
- LB (*Lower Boundary*) atau nilai minimum.
- ALB (*Above the Lower Boundary*) atau nilai diatas batas minimum.
- BUB (*Below the Upper Bound*) atau nilai dibawah batas maksimum.
- UB (*Upper Bound*) atau nilai maksimum
- AUB (*Above the Upper Bound*) atau nilai diatas batas maksimum

Sedangkan untuk *equivalence partitioning* hanya membagi test case menjadi dua bagian yaitu *valid* dan *invalid* data *input* sesuai kriteria tertentu.

### 3. Proses Pengujian

Tahap ini melakukan pengujian pada *test case* yang telah dibuat. Pengujian dilakukan dengan memasukan data *input* berupa *test case* pada sistem, kemudian penguji mencatat hasil *output* yang di amati

### 4. Evaluasi Hasil Pengujian

Tahap ini melakukan proses evaluasi untuk memeriksa keefektifan metode dalam pengujian aplikasi. Evaluasi dapat dilakukan dengan menghitung *defect removal efficiency* yang dapat dilihat pada Persamaan (1).

$$DRE = \frac{\text{jumlah kesalahan yang ditemukan}}{\text{jumlah keseluruhan test case}} \quad (1)$$

## III. HASIL DAN PEMBAHASAN

### 1. Desain Use Case

Beberapa dari desain *use case* adalah sebagai berikut:

#### a. Identifikasi User

*User* atau pengguna dalam aplikasi ini adalah manager pembelian yang bertugas untuk merencanakan jumlah pembelian bahan baku kepada *supplier*.

#### b. Identifikasi Use Case

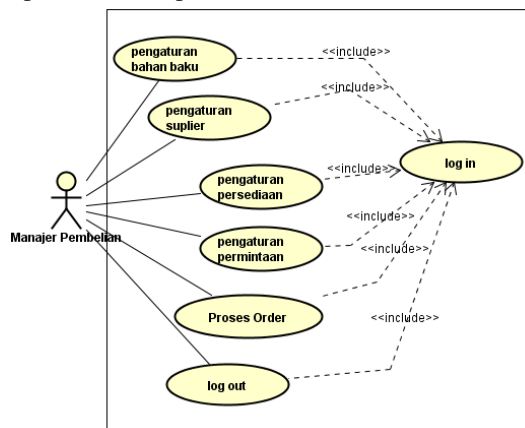
*Use case* mendeskripsikan interaksi antara pengguna dengan aplikasi. Tabel 1 menjelaskan daftar kebutuhan pengguna dan *use case* terhadap aplikasi.

TABEL 1  
DAFTAR USE CASE

No	Daftar Kebutuhan	Use Case
1	Manajer pembelian dapat melakukan <i>log in</i>	<i>Log in</i>
2	Manajer pembelian dapat menambah data, menghapus data, melihat data dan mengedit data bahan baku	Pengaturan Bahan Baku
3	Manajer pembelian dapat menambah data, menghapus data, melihat data dan mengedit data supplier	Pengaturan Suplier
4	Manajer pembelian dapat menambah data, menghapus data, melihat data dan mengedit data persediaan bahan baku	Pengaturan Persediaan Bahan Baku
5	Manajer pembelian dapat melihat dan menambah permintaan	Pengaturan Permintaan
6	Manajer melakukan proses <i>order</i> bahan baku	Proses <i>Order</i>
7	Manajer pembelian dapat <i>logout</i> dari aplikasi	<i>Log out</i>

### c. Desain Use Case Diagram

Berdasarkan Tabel 1. *Use case diagram* dipresentasikan pada Gambar 1.



Gambar 1. *Use Case Diagram*

### 2. Desain Test Case

Tahap yang harus dilakukan dalam mendesain *test case* adalah sebagai berikut :

#### a. Membuat skenariodari *use case diagram*.

Skenario *use case* untuk *log in* dijelaskan pada Tabel 2. dan skenario untuk pengaturan *supplier* dijelaskan Tabel 3.

TABEL 2  
SKENARIO LOGIN

Nama Skenario	<i>Log In</i>
User	Manajer Pembelian
Tujuan	Masuk ke aplikasi
Kondisi Awal	<i>Form log in</i> ditampilkan aplikasi
Kondisi Akhir	<i>Profil</i> pada <i>admin dashboard</i> ditampilkan aplikasi
<b>Skenario Utama :</b>	
1. <i>User</i> diminta aplikasi untuk memasukkan <i>username</i> dan <i>password</i>	
2. <i>username</i> dan <i>password</i> dimasukkan oleh <i>user</i>	
3. <i>User</i> menekan tombol <i>log in</i>	
<b>Skenario Alternatif 1 : <i>Invalid username/ password</i></b>	
1. <i>User</i> diminta aplikasi untuk memasukkan <i>username</i> dan <i>password</i>	

2. *User* tidak mengisi *username/password* atau *user* memasukkan *username/ password* invalid
3. *User* menekan tombol *login*
4. Aplikasi menampilkan pesan kesalahan

TABEL 3  
SKENARIO PENGATURAN *SUPPLIER*

Nama Skenario	Pengaturan <i>Supplier</i>
User	Manajer Pembelian
Tujuan	Mengizinkan manajer pembelian untuk melihat, menambah, mengubah dan menghapus data <i>supplier</i>
Kondisi Awal	<i>User</i> harus <i>log in</i> terlebih dahulu
Kondisi Akhir	Data <i>supplier</i> berhasil dilihat/ ditambah/diedit/dihapus

#### Skenario Utama :

1. Aplikasi meminta *user* memilih aksi untuk melihat/menambah/megubah/menghapus
2. Jika *user* memilih melihat, maka prosedur menampilkan data dijalankan
3. Jika *user* memilih menambah, maka prosedur menambah data dijalankan
4. Jika *user* memilih mengubah, maka prosedur mengubah data dijalankan
5. Jika *user* memilih menghapus, maka prosedur menghapus data dijalankan

#### Skenario Utama 1 : Melihat data *supplier*

1. Aplikasi meminta *user* memasukkan nama *supplier* yang dicari
2. *User* memasukkan nama *supplier* dan menekan tombol cari
3. Aplikasi menampilkan data *supplier*

#### Skenario Utama 2 : Menambah data *supplier*

1. *User* menekan tombol tambah data *supplier*
2. Aplikasi meminta *user* memasukkan data *supplier* berupa nama, alamat dan no hp
3. *User* memasukkan semua data dan menekan tombol simpan
4. Data berhasil ditambahkan ke aplikasi

#### Skenario Utama 3 : Mengubah data *supplier*

1. *User* menekan tombol *edit* pada data *supplier* yang diubah
2. Aplikasi meminta *user* memasukkan data *supplier* yang baru berupa nama, alamat dan no hp
3. *User* memasukkan semua data yang baru dan menekan tombol simpan
4. Data berhasil diubah

#### Skenario Utama 4 : Menghapus data *supplier*

1. *User* menekan tombol hapus pada data *supplier* yang dihapus
2. Aplikasi menampilkan pesan konfirmasi ingin menghapus
3. *User* menyetujui hapus data *supplier* dengan menekan tombol OK
4. Data berhasil dihapus

#### Skenario Alternatif 1 : Data tidak ditemukan

Ketika data *supplier* yang dicari tidak ditemukan maka aplikasi menampilkan pesan kesalahan

#### Skenario Alternatif 2 : *Invalid* data nama/alamat/ no hp untuk tambah dan ubah

1. *User* tidak memasukkan data atau memasukkan data nama/alamat/no hp dengan format yang salah
2. Aplikasi menampilkan pesan kesalahan

#### Skenario Alternatif 3 : *User* membatalkan menambah atau mengubah data

1. *User* mengisi data nama/alamat/no hp
2. *User* tidak menekan tombol simpan tapi menekan tombol Kembali
3. Data baru tidak disimpan di aplikasi

#### Skenario Alternatif 4 : *User* membatalkan menghapus data

1. *User* menekan tombol hapus pada data *supplier* yang dihapus
2. Aplikasi menampilkan pesan konfirmasi ingin menghapus
3. *User* tidak menyetujui hapus data *supplier* dengan menekan tombol batal
4. Data tidak dihapus

b. Membuat *test case matrix*

*Test case* adalah sekumpulan dari data input atau kondisi yang akan diuji dan hasil yang diharapkan. Tampilan halaman *log in* yang dimaksud dipresentasikan pada Gambar 2.

Gambar 2. Halaman *Log in*

*Test case matrix* dibuat dari skenario yang sudah dibuat. Tabel 4 menjelaskan *test case matrix* pada skenario login.

TABEL 4  
TEST CASE MATRIX LOGIN

id	Skenario	Input Username	Input Password	Expect-ed Output
1	Skenario Utama	Valid	Valid	Berhasil Log in
2	Skenario Alternatif 1	Invalid	Valid	Tampil pesan kesalahan-an
3	Skenario Alternatif 1	Valid	Invalid	Tampil pesan kesalahan-an
4	Skenario Alternatif 1	Invalid	Invalid	Tampil pesan kesalahan-an

Tampilan halaman pengaturan *supplier* dipresentasikan pada Gambar 3.

Gambar 3. Halaman Pengaturan *Supplier*

*Test case matrix* skenario pengaturan *supplier* dijelaskan pada Tabel 5.

TABEL 5  
TEST CASE MATRIX PENGATURAN SUPPLIER

No	Skena-rio	Input Nama	Input Alamat	Input No HP	Expect-ed Output
1	Skena-rio Utama 1	Valid	-	-	Tampil data <i>supplier</i>
2	Skena-rio Utama 2	Valid	Valid	Valid	Data berhasil ditam-bah
3	Skena-rio Utama 3	Valid	Valid	Valid	Data berhasil diubah
4	Skena-rio Alter-natif 1	Invalid	-	-	Data berhasil dihapus
5	Skena-rio Alter-natif 2	Valid	Valid	Invalid	Tampil pesan kesa-lahan
6	Skena-rio Alter-natif 2	Valid	Invalid	Valid	Tampil pesan kesa-lahan
7	Skena-rio Alter-natif 2	Valid	Invalid	Invalid	Tampil pesan kesa-lahan
8	Skenario Alternatif 2	Invalid	Valid	Valid	Tampil pesan kesa-lahan
9	Skenario Alternatif 2	Invalid	Valid	Invalid	Tampil pesan kesa-lahan
10	Skenario Alternatif 2	Invalid	Invalid	Valid	Tampil pesan kesa-lahan
11	Skenario Alternatif 2	Invalid	Invalid	Invalid	Tampil pesan kesa-lahan

c. Desain Nilai Test Case

Sebelum mendesain nilai *test case* pada masing-masing data *input*, setiap data *input* harus ditentukan dulu nilainya yang *valid* untuk mempermudah menentukan data *invalid*. *Validity Check* untuk data login disajikan pada Tabel 6 dan *Validity check* untuk data *supplier* disajikan pada Tabel 7.

TABEL 6  
VALIDITY CHECK DATA LOG IN

id	Data Input	Deskripsi
VC1	Username	Username tidak boleh kosong
VC2	Username	Username minimal 6 karakter dan maksimal 12 karakter
VC3	Username	Username hanya terdiri dari huruf, angka dan special karakter tetapi tidak boleh dengan spasi
VC4	Username	Username harus ada di dalam database
VC5	Password	Password tidak boleh kosong
VC6	Password	Password minimal 8 karakter dan maksimal 12 karakter
VC7	Password	Password hanya terdiri dari huruf, angka dan special karakter tetapi tidak boleh dengan spasi
VC8	Password	Password harus ada di dalam database

TABEL 7  
VALIDITY CHECK DATA SUPPLIER

id	Data Input	Deskripsi
VC1	Nama	Nama tidak boleh kosong
VC2	Nama	Hanya terdiri dari huruf dan spasi
VC3	Alamat	alamat tidak boleh kosong
VC4	Alamat	Hanya terdiri dari huruf, angka dan spasi
VC5	No HP	No hp tidak boleh kosong
VC6	No HP	Hanya terdiri dari angka
VC7	No HP	Minimal 9 karakter maksimal 12 karakter

Tabel 8 menjelaskan kelas *Boundary Value Analysis* pada *log in*.

TABEL 8  
KELAS BOUNDARY VALUE ANALYSIS LOGIN

No	Data Input	Nilai	Valid/Invalid	Keterangan
1	User-name	edi12	Invalid	BLB
		edi123	Valid	LB
		edi1234	Valid	ALB
		edipras1234	Valid	BUB
		edipras12345	Valid	UB
		edipras12346	Invalid	AUB
2	Pass-word	admin12	Invalid	BLB
		admin123	Valid	LB
		admin_123	Valid	ALB
		admin_12345	Valid	BUB
		admin_123456	Valid	UB
		admin_1234567	Invalid	AUB

TABEL 9  
KELAS EQUIVALENCE PARTITIONING LOGIN

No	Data Input	Nilai	Valid/Invalid	Keterangan
1	Username	edi123	Valid user-name	Terdapat di Data-base
2	Username		Invalid user-name	Kosong
3	Username	alya123	Invalid user-name	Tidak Terdapat di Data-base
4	Username	edi 123	Invalid user-name	Terdapat spasi
5	Password	admin123	Valid user-name	Terdapat di Data-base
6	Password		Invalid user-name	Kosong
7	Password	admin8899	Invalid user-name	Tidak Terdapat di Data-base
8	Password	admin 123	Invalid user-name	Terdapat spasi

Kelas *Boundary Value Analysis* pada pengaturan *supplier* dapat dilihat pada Tabel 9.

TABEL 9  
KELAS BOUNDARY VALUE ANALYSIS SUPPLIER

No	Data Input	Nilai	Valid/Invalid	Keterangan
1	No HP	03145543	Invalid	BLB
		031455432	Valid	LB
		0314554324	Valid	ALB

		08567423456	Valid	BUB
		085674234564	Valid	UB
		0856742345643	Invalid	AUB

Tabel 10 menjelaskan kelas *Equivalence Partitioning* pada pengaturan *supplier*

TABEL 10  
KELAS EQUIVALENCE PARTITIONING SUPPLIER

No	Data Input	Nilai	Valid/Invalid	Keterangan
1	Nama	Siska Putri	Valid	
2	Nama	Siska123	Invalid	Terdapat angka
3	Nama		Invalid	Kosong
4	Alamat	JL Bali No 24 Malang	Valid	
5	Alamat		Invalid	Kosong
6	No HP	031455432	Valid	
7	No HP		Invalid	Kosong
8	No HP	abc123	Invalid	Terdapat huruf

Desain nilai *test case* pada *log in* dijelaskan Tabel 11.

TABEL 11  
KELAS EQUIVALENCE PARTITIONING SUPPLIER

Id TC	Username	Password	Expected Output
TC1	edi123	admin123	Berhasil log in
TC2	edi12	admin123	Invalid user-name
TC2	edipras12346	admin123	Invalid user-name
TC3		admin123	Username kosong
TC4	alya123	admin123	Username tidak ditemukan
TC5	edi 123	admin123	Invalid user-name
TC6	edi123	admin12	Invalid password
TC7	edi123	admin_1234567	Invalid Password
TC8	edi123		Password kosong
TC9	edi123	admin8899	Password tidak ditemukan
TC 10	edi123	admin 123	Invalid Password

Desain nilai *test case* pada pengaturan *supplier* dijelaskan pada Tabel 12.

TABEL 12  
TEST CASE PENGATURAN

Id TC	Nama	Alamat	No HP	Expected Output
TC 11	Siska Putri	JL Bali No 24 Malang	031455432	Berhasil disimpan
TC 12	Siska123	JL Bali No 24 Malang	031455432	Invalid Nama
TC 13		JL Bali No 24 Malang	031455432	Nama Kosong
TC 14	Siska Putri		031455432	Alamat Kosong
TC 15	Siska Putri	L Bali No 24 Malang	03145543	Invalid No HP

<b>ID TC</b>	<b>Nama</b>	<b>Alamat</b>	<b>No HP</b>	<b>Expected Output</b>
TC 16	Siska Putri	L Bali No 24 Malang	08567423 45643	Invalid No HP
TC 17	Siska Putri	L Bali No 24 Malang		No HP Kosong
TC 18	Siska Putri	L Bali No 24 Malang	abc123	Invalid No HP

### 3. Proses Pengujian

Hasil pengujian *test case* Tabel 11. pada halaman *log in* dijelaskan pada Tabel 13.

TABEL 13  
HASIL PENGUJIAN TEST CASE LOG IN

<b>ID Test Case</b>	<b>Expected Output</b>	<b>Observed Output</b>	<b>Pass/Fail</b>
TC1	Berhasil login	Berhasil login	Pass
TC2	Invalid data username	Valid data username	Fail
TC3	Invalid data username	Valid data username	Fail
TC4	Username kosong	Username kosong	Pass
TC5	Username tidak ditemukan	Username tidak ditemukan	Pass
TC6	Invalid data username	Valid data username	Fail
TC7	Invalid data password	Valid data password	Fail
TC8	Password kosong	Password kosong	Pass
TC9	Password tidak ditemukan	Password tidak ditemukan	Pass
TC10	Invalid data Password	Valid data Password	Fail

Hasil pengujian *test case* Tabel 12. pada halaman *supplier* dijelaskan pada Tabel 14.

TABEL 14  
HASIL PENGUJIAN TEST CASE SUPPLIER

<b>ID Test Case</b>	<b>Expected Output</b>	<b>Observed Output</b>	<b>Pass/Fail</b>
TC11	Berhasil disimpan	Berhasil disimpan	Pass
TC12	Invalid Nama	Valid Nama	Fail
TC13	Nama Kosong	Nama Kosong	Pass
TC14	Alamat Kosong	Alamat Kosong	Pass
TC15	Invalid No HP	Valid No HP	Fail
TC16	Invalid No HP	Valid No HP	Fail
TC17	No HP Kosong	No HP Kosong	Pass
TC18	Invalid No HP	Invalid No HP	Pass
TC19	Berhasil disimpan	Berhasil disimpan	Pass
TC20	Invalid Nama	Valid Nama	Fail

### 4. Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian pada Tabel 13. dan Tabel 14. dari total 20 Test Case, 11 Test case pass atau lolos uji sedangkan 9 *test case* memberikan hasil *fail* atau tidak lolos uji. Sehingga didapatkan nilai DRE sebesar 0.45 atau 45%. Hal ini menunjukkan metode ini sangat efektif dalam menemukan kesalahan dalam aplikasi.

## IV. KESIMPULAN

Penggabungan metode *Boundary Value Analysis* dan *Equivalence Partitioning* dengan pendekatan *use case* untuk desain *test case* pada metode Black Box Testing sangat efisien dalam menemukan kesalahan aplikasi. Hal ini ditunjukkan nilai DRE 0.45 yang berarti 45% *test case* yang dibangun tidak lolos uji atau dapat menemukan

kesalahan dari aplikasi. Saat ini pengujian masih dilakukan dari sisi fungsional aplikasi, maka untuk penelitian selanjutnya pengujian bisa dilakukan dari sisi non fungsional aplikasi. Selain itu bisa melakukan komparasi dengan metode *Black Box Testing* yang lainnya.

## V. UCAPAN TERIMA KASIH

Penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu terlaksanannya penelitian ini. Ucapan terima kasih setinggi tinggi nya kepada Departement Vokasi Universitas Brawijaya Malang dan LPPM Universitas Merdeka Malang yang telah memfasilitasi Kerjasama penelitian dosen antar perguruan tinggi.

## REFERENSI

- [1] G. D. Everett, *Software Testing Software Development Life Cycle*. Canada: IEEE Press, 2007.
- [2] J. Tian, *Software Quality Engineering*. Canada: Wiley, 2005.
- [3] P. Ron, *Software testing*. United States of America: SAMS Publishing, 2004.
- [4] G. Fournier, *Essential Software Testing A Use-Case Approach*. New York: CRC Press, 2009.
- [5] S. Jagtap, V. Gawade, R. Pawar, S. Shendge, and P. Avhad, "Generate Test Cases From UML Use Case and State Chart Diagrams," *Int. Res. J. Eng. Technol.*, vol. 3, no. 10, pp. 873–881, 2016, [Online]. Available: [www.irjet.net](http://www.irjet.net).
- [6] S. K. Swain, D. P. Mohapatra, and R. Mall, "Test Case Generation Based on Use case and Sequence Diagram," *Int. J. Softw. Eng.*, no. JANUARY, pp. 21–52, 2010.
- [7] G. J. Myers, T. Badgett, and C. Sandle, *The Art of Software Testing*. Canada: Wiley, 2012.
- [8] K. Naik and P. Tripathy, *Software Testing and Quality Assurance : Theory and Practice*. Canada: Wiley, 2008.
- [9] I. P. A. Prayudha, R. S. Hartati, and Y. Divayana, "Boundary Value Analysis Testing Techniques on Learning Management System Applications," *Int. J. Eng. Emerg. Technol.*, vol. 4, no. 2, pp. 25–28, 2019.
- [10] D. Andriansyah, "Pengujian Kotak Hitam Boundary Value Analysis Pada Sistem Informasi Manajemen Konseling Tugas Akhir," *Indones. J. Netw. Secur.*, vol. 7, no. 1, pp. 20–25, 2018, [Online]. Available: <http://ijns.org/journal/index.php/ijns/article/view/1496>.
- [11] R. Pramudita, "Pengujian Black Box pada Aplikasi Pertokoan Minimarket Menggunakan Metode Equivalence Partitioning," *Informatics Educ. Profesiionals*, vol. 4, no. 2, pp. 193–202, 2020, doi: 10.32493/jtsi.v3i2.4695.
- [12] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, "Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, pp. 155–161, 2019, doi: 10.32493/jtsi.v2i4.3771.
- [13] M. E. Khan and F. Khan, "A Comparative Study of White Box , Black Box and Grey Box Testing Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 12–15, 2012.
- [14] S. Ikhlashi and H. P. Putro, "Komparasi Dua Teknik Black Box Testing: Equivalence Partitioning dan Boundary Value Analysis," *Annu. Res. Semin. 2019*, vol. 5, no. 1, pp. 213–220, 2019.
- [15] Y. Singh, *Software Testing*. India: Cambridge University Press, 2012.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

