

Automation Testing Terhadap E-recruitment PT. ABC Menggunakan Tools Serenity BDD Dengan Teknik Equivalence Partitions dan Boundary Value Analysis (BVA)

Rizania Fayza Indhira¹, Dwi Fatrianto Suyatno²

^{1,2} Sistem Informasi, Fakultas Teknik, Universitas Negeri Surabaya

rizania.20039@mhs.unesa.ac.id

dwifatrianto@unesa.ac.id

Abstrak— *E-recruitment* adalah *website* perekrutan karyawan baru yang saat ini sudah diterapkan oleh PT. ABC dimana membantu dalam seluruh proses manajemen dan administrasi. Namun, *E-recruitment* PT. ABC ini dalam pengujiannya masih dilakukan secara manual yang memungkinkan ditemukannya kesalahan yang terlewat akibat *human error*, sehingga nantinya akan berdampak negatif bagi reputasi perusahaan. Karena itu, perlu dilakukannya *automation testing* yang menggunakan *tools* Serenity BDD menerapkan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) dalam pembuatan *test case* untuk mengetahui fungsionalitas dari *E-recruitment* PT. ABC, memberikan rekomendasi perbaikan jika ditemukan kesalahan, serta melakukan perbandingan antara teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA). Proses *automation testing* pada *E-recruitment* PT. ABC dimulai dengan membuat daftar *field* yang akan diuji, menyusun *test case*, kemudian dilanjutkan dengan pembuatan file *pom.xml*, file *cucumber test suite*, file *serenity.properties*, file *serenity.conf*, file *feature*, *class pages*, *class step definitions*, lalu menjalankan *automation testing* dengan memasukkan perintah *mvn clean verify*. Hasil penelitian ini menunjukkan bahwa fungsionalitas *E-recruitment* PT. ABC cukup baik dengan persentase sebesar 62% dari total *test case* sebanyak 709 butir dengan 442 butir berstatus *pass* (berhasil) dan 267 butir lainnya berstatus *fail* (gagal). Rekomendasi perbaikan akan diberikan pada 267 butir yang memiliki status *fail* tersebut. Terdapat beberapa perbandingan antara teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) mulai dari sisi hasil *automation testing*, *field*, dan data uji.

Kata Kunci— *E-recruitment*, *Automation Testing*, Serenity BDD, *Equivalence Partitions*, *Boundary Value Analysis* (BVA)

I. PENDAHULUAN

Selama beberapa tahun terakhir, sebanyak 93% dari 500 perusahaan internasional yang telah menerapkan *E-recruitment* yang juga diikuti oleh perusahaan dalam negeri dalam merekrut karyawan baru mereka [1]. *E-recruitment* atau “Electronic Recruitment” merupakan proses perekrutan karyawan baru pada perusahaan yang memanfaatkan media elektronik yang dapat membantu dalam manajemen seluruh proses perekrutan karyawan baru, mulai dari pengiklanan info lowongan pekerjaan, proses seleksi, menghemat biaya, hingga proses pelaporan [2]. *E-recruitment* ini lebih banyak tersedia pada *platform website* karena minat masyarakat yang tinggi terhadap *platform website* dalam kehidupan sehari-hari. Oleh karena itu, proses pengujian terhadap perangkat lunak

terutama *platform website* adalah proses penting dalam menjaga kualitas dan memastikan fungsionalitas dari fitur-fitur yang ada didalamnya sudah berjalan sesuai yang diharapkan atau belum [3]. Pengujian ini perlu dilakukan guna memeriksa dan mencegah kesalahan yang bisa saja muncul, sehingga kesalahan tersebut dapat dikurangi atau dihapus setelah perangkat lunak diluncurkan ke publik [4].

Metode yang dapat digunakan untuk pengujian pada perangkat lunak yakni *Black Box Testing* yang memiliki fokus untuk menguji fungsionalitas antarmuka perangkat lunak tanpa menghiraukan kode penyusunnya [5]. *Black Box Testing* ini dilakukan pada sudut pandang pengguna dimana cara kerjanya adalah dengan memasukkan inputan dan mengamati keluaran apakah sudah sesuai dengan yang diharapkan atau belum. Teknik dalam metode *Black Box Testing* contohnya yaitu teknik *Equivalence Partitions* yang membagi data uji menjadi kategori valid dan invalid serta teknik *Boundary Value Analysis* (BVA) yang data ujinya dibagi berdasarkan batas bawah ($x-1$), nilai batas (x), dan batas atas ($x+1$) [6].

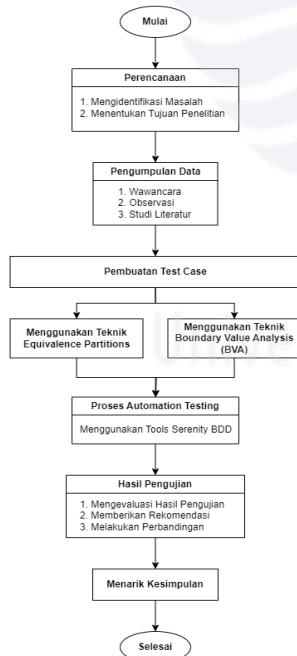
Pengujian pada perangkat lunak dapat dilakukan secara manual maupun otomatis. Selama ini, proses pengujian masih dilakukan secara manual yang masih terdapat kekurangan dalam pelaksanaannya mulai dari membutuhkan waktu yang lebih lama hingga masih ditemukannya kesalahan akibat *human error*, sehingga dapat memberikan dampak negatif bagi reputasi perusahaan [7]. Pengujian otomatis atau *automation testing* masih dilakukan kurang dari 20% padahal memiliki lebih banyak keuntungan ketimbang pengujian manual [8]. *Automation testing* merupakan pengujian terhadap perangkat lunak yang memanfaatkan skrip atau *tools* yang dapat menemukan kesalahan lebih dini dengan akurasi tinggi, meningkatkan efisiensi dan cakupan pengujian, menghemat waktu, sumber daya, serta biaya [9]. Salah satu *tools* yang populer untuk melakukan *automation testing* adalah Serenity BDD. Serenity BDD merupakan *library open source* dalam melakukan *automation testing* yang menghasilkan *report* untuk menganalisis kualitas dari perangkat lunak yang dilihat dari berapa banyak fitur berhasil dan gagal beserta persentasenya hingga *response time* dari setiap *steps* yang dijalankan [7]. Serenity BDD ini dapat digunakan untuk *automation testing* pada berbagai *platform* mulai dari *website*, API, hingga *mobile* serta pada sistem operasi Windows dan macOS.

PT. ABC merupakan perusahaan manufaktur yang bergerak di bidang kosmetik dan toiletris. Saat ini PT. ABC sudah

menerapkan *E-recruitment* yang berbasis *platform website* dalam perekrutan karyawan baru. Akan tetapi, proses pengujian yang dilakukan masih secara manual dengan memasukkan inputan dan mengamati validasi dari hasil keluaran. Hal tersebut memungkinkan masih adanya kesalahan yang ditemukan karena terlewat akibat *human error*, sehingga akan berdampak negatif bagi reputasi perusahaan nantinya karena *E-recruitment* ini akan diakses oleh banyak orang khususnya para pencari kerja. Oleh karena itu, diperlukannya *automation testing* pada *E-recruitment* PT. ABC menggunakan *tools* Serenity BDD yang menggunakan IntelliJ IDEA sebagai *text editor tools* dengan metode *Black Box Testing* dimana teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) digunakan dalam pembuatan *test casenya* untuk mengetahui fungsionalitas dari *E-recruitment* dan mengevaluasi hasil pengujian tersebut, sehingga dapat memberikan rekomendasi perbaikan kesalahan yang menjadi masukan positif bagi perusahaan. Tak hanya itu, *automation testing* ini juga dilakukan untuk mengetahui perbandingan antara teknik *Equivalence Partitions* dan teknik *Boundary Value Analysis* (BVA).

II. METODE PENELITIAN

Penelitian ini menggunakan *E-recruitment* PT. ABC sebagai objek penelitian dengan jenis penelitian kualitatif yang berdasarkan data sesuai kejadian di lapangan. *Automation testing tools* yang digunakan yakni Serenity BDD dengan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) sebagai teknik pengujian sistem. Alur penelitian dapat dilihat seperti berikut ini.



Gbr. 1 Alur Penelitian

A. Perencanaan

Tahap ini dimulai dengan mengidentifikasi masalah dari *E-recruitment* PT. ABC dengan melakukan wawancara bersama narasumber PT. ABC yang kemudian dilanjutkan dengan menentukan tujuan penelitian dari masalah yang sudah diidentifikasi itu.

B. Pengumpulan Data

Data yang akan digunakan pada penelitian ini didapatkan dari melakukan wawancara bersama narasumber PT. ABC mengenai objek penelitian secara lebih mendalam, melakukan observasi secara langsung pada *website E-recruitment* PT. ABC yang nantinya akan didapatkan daftar *field* mana saja untuk pedoman dalam membuat *test case*, hingga melakukan studi literatur dengan membaca penelitian sebelumnya yang memiliki topik serupa sebagai bahan acuan bisa berasal dari jurnal, buku, atau artikel.

C. Pembuatan Test Case

Tahapan sebelum proses *automation testing* adalah pembuatan *test case* yang dibedakan berdasarkan tekniknya yaitu yang menggunakan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) dimana akan dijadikan acuan dalam *automation testing* pada *E-recruitment* PT. ABC. Fitur yang akan diuji mulai dari *register*, *login*, lupa password, profil, lowongan, lowongan disimpan, lamaran saya, dan juga permintaan pekerjaan dengan total sebanyak 204 *field*. Terdapat total 247 butir *test case* yang menerapkan teknik *Equivalence Partitions*. Berikut ini adalah beberapa daftar *test case* yang menggunakan teknik *Equivalence Partitions*.

TABEL I
TEST CASE TEKNIK EQUIVALENCE PARTITIONS

Test Case ID	Test Feature	Test Type	Test Case Description	Expected Result
TC-107	Login	Valid	User memasukkan semua field dengan benar	User melihat pop up berhasil login
TC-130	Profil	Invalid	User memasukkan nomor KTP dengan format alphabet	User melihat pesan peringatan "The KTP number is invalid"
TC-244	Permintaan pekerjaan	Invalid	User mengosongkan posisi pekerjaan	User melihat pesan peringatan "The job position field is required"

Sedangkan, total *test case* yang akan diuji dengan menerapkan teknik *Boundary Value Analysis* (BVA) sebanyak 462 butir, berikut ini adalah beberapa daftar *test casenya*.

TABEL II
TEST CASE TEKNIK BOUNDARY VALUE ANALYSIS

Test Case ID	Test Feature	Test Type	Test Case Description	Expected Result
--------------	--------------	-----------	-----------------------	-----------------

Test Case ID	Test Feature	Test Type	Test Case Description	Expected Result
TC-459	Login	Nilai batas	User memasukkan email 29 karakter	User melihat pop up berhasil login
TC-482	Profil	Nilai batas bawah	User memasukkan nomor KTP 15 karakter	User melihat pesan peringatan "The KTP number is invalid"
TC-703	Permintaan pekerjaan	Nilai batas atas	User memasukkan posisi pekerjaan 201 karakter	User melihat pesan peringatan "You have exceeded the limit of 200 characters"

```
package starter;
import io.cucumber.junit.CucumberOptions;
import net.serenitybdd.cucumber.CucumberWithSerenity;
import org.junit.runner.RunWith;

@RunWith(CucumberWithSerenity.class)
@CucumberOptions(
    plugin = {"pretty"},
    features = "src/test/resources/features"
)

public class CucumberTestSuite {
}
```

Gbr. 3 File cucumber test suite

3) Membuat file *serenity.properties*

Kemudian membuat file *serenity.properties* yang berisi dari base url website E-recruitment PT. ABC, jenis, dan lokasi dari *webdriver*.

```
webdriver.driver=chrome
webdriver.chrome.driver=src/test/resources/webdriver/chromedriver.exe
webdriver.base.url= https://vicicareer.vci.co.id/
```

Gbr. 4 File *serenity.properties*

4) Membuat file *serenity.conf*

Lalu dilanjutkan dengan membuat file *serenity.conf* yang berisi *driver*, waktu *timeout*, serta cara pengambilan *screenshots* dari setiap *step* yang dilakukan selama proses *automation testing*.

```
serenity {
    take_screenshots = FOR_FAILURES
    take_screenshots = FOR_EACH_ACTION
}

headless_mode = false

webdriver {
    driver = chrome
    capabilities {
        browserName = "chrome"
        acceptInsecureCerts = true
        "goog:chromeOptions" {
            args = ["remote-allow-origins=*", "test-type", "no-sandbox", "ignore-certificate-errors", "start-maximized", "incognito", "disable-infobars", "disable-gpu", "disable-default-apps", "disable-popup-blocking"]
        }
    }
    wait {
        for {
            timeout = 30000
        }
    }
}
```

Gbr. 5 File *serenity.conf*

5) Membuat file *feature*

Langkah selanjutnya yakni membuat file *feature* yang berisi skenario dari setiap *step* selama *automation testing*, ditulis dengan menggunakan bahasa inggris yang mengimplementasikan bahasa *Gherkin* yaitu *Given* (kondisi awal sistem sebelum pengujian), *When* (tindakan yang dilakukan selama pengujian), *Then* (hasil sebenarnya dari pengujian yang dilakukan) [10].

D. Proses Automation Testing

Automation testing terhadap E-recruitment PT. ABC dilakukan setelah pembuatan *test case* selesai. *Automation testing* ini menggunakan tools Serenity BDD dengan bantuan IntelliJ IDEA sebagai *text editor tools* serta *web browser* Chrome berdasarkan dengan *test case* yang sudah dibuat sebelumnya dimana nantinya akan menghasilkan *report test case* mana saja yang berhasil dan gagal. Langkah-langkah yang dilakukan saat proses *automation testing* menggunakan tools Serenity BDD yakni sebagai berikut:

1) Membuat file *pom.xml*

Langkah pertama yakni membuat file *pom.xml* yang berisi *properties*, *dependencies*, dan *plugin* yang dibutuhkan untuk proses *automation testing* nantinya.

```
<properties>
<maven.compiler.source>8</maven.compiler.source>
<maven.compiler.target>8</maven.compiler.target>
<serenity.version>3.6.21</serenity.version>
</properties>
<dependencies>
<dependency>
<groupId>net.serenity-bdd</groupId>
<artifactId>serenity-core</artifactId>
<version>${serenity.version}</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>net.serenity-bdd</groupId>
<artifactId>serenity-cucumber</artifactId>
<version>${serenity.version}</version>
<scope>test</scope>
</dependency>
</dependencies>
```

Gbr. 2 File *pom.xml*

2) Membuat file *cucumber test suite*

File *cucumber test suite* ini berisi jenis *plugin* dan lokasi dari file *feature*. Pembuatan file *cucumber test suite* ini merupakan langkah kedua dari proses *automation testing*.

```

Feature: Login
  As a user i want to login so that i can see login success pop up

Background:
  Given user on home page
  And user click profile icon
  And user on login page

@TC-107
Scenario: as a user i success login with input valid email and valid password
  When user input valid email for login
  And user input valid password for login
  And user click login button
  Then user see login success pop up

@TC-108
Scenario: as a user i failed login with input empty email
  When user input empty email for login
  And user input valid password for login
  And user click login button
  Then user see please fill out this field alert
    
```

Gbr. 6 File feature

```

import starter.pages.DashboardPage;
import starter.pages.LoginPage;

public class LoginSteps {
    @Steps
    LoginPage loginPage;
    DashboardPage dashboardPage;

    @When("user input valid email for login")
    public void userInputValidEmailForLogin() {
        loginPage.inputValidEmailForLogin("rizania.20039@mhs.unesa.ac.id");
    }

    @And("user input valid password for login")
    public void userInputValidPasswordForLogin() {
        loginPage.inputValidPasswordForLogin("Indhira#25");
    }

    @And("user click login button")
    public void userClickLoginButton() {
        loginPage.clickLoginButton();
    }
}
    
```

Gbr. 8 Class step definitions

6) Membuat class pages

Selanjutnya membuat *class pages* yang berisi detail elemen dari halaman *website E-recruitment PT. ABC* yang memanfaatkan perintah *By* dan aksi dari setiap *step* yang dilakukan dengan memanggil elemen yang sudah didefinisikan tersebut.

```

package starter.pages;

import net.serenitybdd.core.pages.PageObject;
import net.serenitybdd.screenplay.actions.Switch;
import net.thucyides.core.annotations.Step;
import org.openqa.selenium.By;

public class LoginPage extends PageObject {
    private By loginPageTitle(){
        return By.xpath("//h3[text()='Login']");
    }

    private By emailForLoginField(){
        return By.xpath("//*[@id='formLogin']/div[1]/input");
    }

    private By loginButton(){
        return By.id("loginButton");
    }

    @Step
    public void onLoginPage(){
        $(loginPageTitle()).isDisplayed();
    }

    @Step
    public void inputValidEmailForLogin(String validEmail){
        $(emailForLoginField()).type(validEmail);
    }
}
    
```

Gbr. 7 Class pages

7) Membuat class step definitions

Dilanjutkan dengan membuat *class step definitions* yang menjadi jembatan antara *file feature* dan *class steps* dimana akan dimasukkan juga inputan dari setiap *field*, sehingga nantinya akan memudahkan ketika ingin menggunakan kembali *step* yang sama untuk pengujian selanjutnya. *Class step definitions* ini terdiri dari *@Given*, *@When*, dan *@Then* tergantung dari *step* yang dibuat.

8) Menjalankan automation testing

Langkah terakhir yakni menjalankan *automation testing* dengan memasukkan perintah *mvn clean verify* di terminal dimana nantinya setiap *step* akan dijalankan satu-persatu hingga selesai, lalu akan muncul *report automation testing*.

```

Terminal: Local x Command Prompt x + v
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sasus\IdeaProjects\skripsi_vici>mvn clean verify
    
```

Gbr. 9 Perintah mvn clean verify

E. Hasil Pengujian

Setelah proses *automation testing* terhadap *E-recruitment PT. ABC* menggunakan *tools Serenity BDD* selesai, akan didapatkan hasil pengujian yang akan dievaluasi mana saja *test case* yang berhasil dan gagal. Rekomendasi perbaikan akan diberikan pada *test case* yang masih gagal, kemudian dilakukan perbandingan antara teknik *Equivalence Partitions* dan teknik *Boundary Value Analysis (BVA)* berdasarkan hasil pengujian.

F. Menarik Kesimpulan

Menarik kesimpulan merupakan tahapan terakhir setelah semua proses selesai dilakukan dimana akan berisi bagaimana fungsionalitas dari *E-recruitment PT. ABC*, rekomendasi perbaikan kesalahan, hingga perbandingan antara teknik *Equivalence Partitions* dan *Boundary Value Analysis (BVA)*.

III. HASIL DAN PEMBAHASAN

A. Hasil Automation Testing dengan Teknik Equivalence Partitions

Setelah dilakukan *automation testing* pada *E-recruitment PT. ABC* menggunakan *tools Serenity BDD*, maka akan didapatkan hasil pengujian. Berikut ini adalah beberapa hasil *automation testing* yang menggunakan Teknik *Equivalence Partitions*.

TABEL III
HASIL AUTOMATION TESTING TEKNIK EQUIVALENCE PARTITIONS

Test Case ID	Test Feature	Test Case Description	Actual Result	Status
TC-107	Login	User memasukkan semua field dengan benar	Sistem berhasil menampilkan pop up berhasil login	Pass
TC-130	Profil	User memasukkan nomor KTP dengan format alphabet	User melihat pesan peringatan "The KTP number is invalid"	Fail
TC-244	Permintaan pekerjaan	User mengosongkan posisi pekerjaan	User melihat pesan peringatan "The job position field is required"	Pass

Total test case dengan teknik *Equivalence Partitions* yang dilakukan *automation testing* menggunakan *tools* Serenity BDD yakni sebanyak 247 butir. Berikut ini adalah hasil *automation testing* per fitur.

TABEL IV
HASIL PENGUJIAN TEKNIK EQUIVALENCE PARTITIONS

No.	Fitur	Total Test Case	Status		Tingkat Kesuksesan
			Pass	Fail	
1.	Register	106	49	57	46%
2.	Login	6	6	-	100%
3.	Lupa password	4	4	-	100%
4.	Profil	110	65	45	59%
5.	Lowongan	12	10	2	83%
6.	Lowongan disimpan	2	2	-	100%
7.	Lamaran saya	2	2	-	100%
8.	Permintaan pekerjaan	5	5	-	100%
Total		247	143	104	58%

Dapat dilihat pada tabel 4, hasil *automation testing* pada *E-recruitment* PT. ABC menggunakan teknik *Equivalence Partitions* dengan total 247 butir didapatkan bahwa sebanyak 143 butir berstatus *pass* (berhasil) dan 104 butir lainnya *fail* (gagal) dengan tingkat kesuksesan mencapai 58%, sehingga tingkat fungsionalitasnya cukup baik. Berikut adalah salah satu *report automation testing* dari TC-107 yang memiliki status *pass* (berhasil) dimana didalamnya akan dijelaskan beberapa *step* yang berhasil dilakukan selama proses pengujian yang disertai dengan waktu respon dari setiap *step* tersebut.



Gbr. 10 Report Automation Testing TC-107

B. Hasil Automation Testing dengan Teknik Boundary Value Analysis (BVA)

Hasil *automation testing* yang didapatkan pada *E-recruitment* PT. ABC yang menggunakan *tools* Serenity BDD dengan menerapkan teknik *Boundary Value Analysis* (BVA) yakni sebagai berikut.

TABEL V
HASIL AUTOMATION TESTING TEKNIK BOUNDARY VALUE ANALYSIS (BVA)

Test Case ID	Test Feature	Test Case Description	Actual Result	Status
TC-459	Login	User memasukkan email 29 karakter	Sistem berhasil menampilkan pop up berhasil login	Pass
TC-482	Profil	User memasukkan nomor KTP 15 karakter	User melihat pesan peringatan "The KTP number is invalid"	Fail
TC-703	Permintaan pekerjaan	User memasukkan posisi pekerjaan 201 karakter	User melihat pesan peringatan "You have exceeded the limit of 200 characters"	Fail

Total *automation testing* yang dilakukan pada *E-recruitment* PT. ABC yang menggunakan teknik *Boundary Value Analysis* (BVA) yakni sebanyak 462 butir *test case*. Berikut ini adalah hasil *automation testing* per fitur.

TABEL VI
 HASIL PENGUJIAN TEKNIK BOUNDARY VALUE ANALYSIS (BVA)

No.	Fitur	Total Test Case	Status		Tingkat Kesuksesan
			Pass	Fail	
1.	Register	210	127	83	60%
2.	Login	6	6	-	100%
3.	Lupa password	3	3	-	100%
4.	Profil	222	146	76	66%
5.	Lowongan	6	4	2	67%
6.	Lowongan disimpan	3	3	-	100%
7.	Lamaran saya	3	3	-	100%
8.	Permintaan pekerjaan	9	7	2	78%
Total		462	299	163	65%

Dapat dilihat pada tabel 6, didapatkan hasil *automation testing* terhadap *E-recruitment* PT. ABC yang cukup baik dengan persentase mencapai 65% dimana dari total 462 butir *test case* yang diuji, sebanyak 299 butir berstatus *pass* (berhasil) dan 163 butir lainnya memiliki status *fail* (gagal). Berikut *report automation testing* pada TC-703 yang berstatus *fail* (gagal) dimana akan dijelaskan pada *step* mana yang gagal selama proses pengujian disertai dengan penyebabnya.

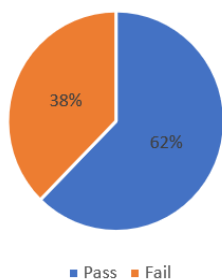


Gbr. 11 Report Automation Testing TC-703

C. Evaluasi Hasil Automation Testing

Setelah dilakukan proses *automation testing* terhadap *E-recruitment* PT. ABC menggunakan tools Serenity BDD dengan menerapkan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA), maka didapatkan hasil tingkat fungsionalitas yang cukup baik dengan persentase sebesar 62% dari total *test case* sebanyak 709 butir dengan rincian yakni 442 butir berstatus *pass* (berhasil) dan 267 butir berstatus *fail* (gagal). Test case yang statusnya gagal tersebut karena *error handling* yang belum diatur saat diberikan masukan yang ekstrim atau tidak terduga, *padahal error handling* ini penting untuk dapat memberikan pesan peringatan yang membantu pengguna mengerti kesalahan yang muncul dan segera memperbaikinya serta memastikan sistem telah berjalan dengan benar. Berikut ini adalah grafik tingkat fungsionalitasnya.

Tingkat Fungsionalitas



Gbr. 12 Grafik Tingkat Fungsionalitas

D. Rekomendasi Perbaikan

Masih terdapat *test case* yang berstatus *fail* (gagal) setelah dilakukan proses *automation testing* terhadap *E-recruitment* PT. ABC menggunakan *tools* Serenity BDD dengan menerapkan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) yakni sebanyak 267 butir. *Test case* yang *fail* tersebut berarti hasil sebenarnya tidak sama dengan hasil yang diharapkan. Oleh karena itu, akan diberikan rekomendasi perbaikan yang diharapkan dapat meningkatkan tingkat fungsionalitas *E-recruitment* PT. ABC tersebut. Berikut ini adalah beberapa rekomendasi perbaikannya.

TABEL VII
REKOMENDASI PERBAIKAN

Test Case ID	Test Feature	Test Case Description	Actual Result	Recommendation
TC-130	Profil	User memasukkan nomor KTP dengan format alphabet	Sistem gagal menampilkan peringatan "The KTP number is invalid"	Menampilkan peringatan "The KTP number is invalid"
TC-482		User memasukkan nomor KTP 15 karakter	Sistem gagal menampilkan pesan peringatan "The KTP number is invalid"	Menampilkan pesan peringatan "The KTP number is invalid"
TC-703	Permintaan pekerjaan	User memasukkan posisi pekerjaan 201 karakter	Sistem gagal menampilkan pesan peringatan "You have exceeded the limit of 200 characters"	Menampilkan pesan peringatan "You have exceeded the limit of 200 characters"

E. Perbandingan antara Teknik *Equivalence Partitions* dan Teknik *Boundary Value Analysis* (BVA)

Didapatkan beberapa perbandingan antara teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA) setelah dilakukan proses *automation testing* yang didapatkan

pada *E-recruitment* PT. ABC menggunakan *tools* Serenity BDD dapat dilihat sebagai berikut.

TABEL VIII
PERBANDINGAN TEKNIK *EQUIVALENCE PARTITIONS* DAN *BOUNDARY VALUE ANALYSIS* (BVA)

No.	Pembanding	Teknik <i>Equivalence Partitions</i>	Teknik <i>Boundary Value Analysis</i> (BVA)
1.	Hasil <i>automation testing</i>	Sebanyak 143 butir <i>test case</i> berhasil dan 104 butir gagal dari total 247 butir dengan persentase 58%	Sebanyak 299 butir <i>test case</i> berhasil dan 163 butir gagal dari total 462 butir dengan persentase 65%
2.	Field	Dapat menguji semua jenis field (input text, date picker, select box, dll)	Hanya dapat digunakan untuk menguji field yang bertipe input text
		Jangkauannya lebih luas dan bebas, sehingga dapat menguji semua kemungkinan yang ada	Hanya menguji field yang memiliki batasan nilai, sehingga cocok untuk sistem yang berfokus pada batasan input nilai
3.	Data uji	Membagi data uji menjadi partisi valid dan invalid	Membagi data uji berdasarkan batas bawah (x-1), nilai batas (x), dan batas atas (x+1)
		Pengujian akan terasa lengkap dan bervariasi karena partisi tipe invalid akan terdiri dari <i>test case</i> dengan banyak kemungkinan	Pengujiannya akan terasa lebih jelas karena hanya menguji berdasarkan batas bawah, nilai batas, dan batas atas

IV. KESIMPULAN

Setelah dilakukannya proses *automation testing* pada *E-recruitment* PT. ABC terdapat beberapa poin kesimpulan yang didapatkan, yaitu

1. Hasil *automation testing* menunjukkan sebanyak 442 butir *test case* berstatus *pass* (berhasil) dan 267 butir berstatus *fail* (gagal) dari total 709 butir *test case* dengan tingkat fungsionalitas *E-recruitment* PT. ABC yang cukup baik dimana persentasenya sebesar 62%.
2. Terdapat beberapa rekomendasi untuk memperbaiki 267 butir *test case* yang berstatus gagal baik itu yang menggunakan teknik *Equivalence Partitions* dan *Boundary Value Analysis* (BVA), salah satu contohnya yakni pada TC-482 yang dimana user memasukkan nomor KTP 15 karakter, sebaiknya sistem dapat menampilkan pesan peringatan "The

KTP number is invalid” karena nomor KTP yang terdiri dari 16 karakter.

3. Perbandingan antara teknik *Equivalence Partitions* dengan teknik *Boundary Value Analysis* (BVA) dilihat dari hasil *automation testing* yakni teknik *Equivalence Partitions* menunjukkan persentase 58% dengan 143 butir *test case* berhasil dan 104 butir gagal dari total 247 butir, sedangkan teknik *Boundary Value Analysis* (BVA) menunjukkan sebanyak 299 butir *test case* berhasil dan 163 butir gagal dari total 462 butir dengan persentase 65%. Kemudian teknik *Equivalence Partitions* ini menguji semua jenis *field* dengan data uji yang dibagi menjadi valid dan invalid, sehingga hasil pengujian akan terasa lengkap dan bervariasi karena jangkauan yang luas. Sedangkan teknik *Boundary Value Analysis* (BVA) hanya dapat menguji *field* yang berupa *input text* dengan data uji yang memiliki batasan nilai, sehingga hasil pengujiannya akan lebih jelas, terarah, dan cocok untuk sistem yang fokusnya pada batas nilai inputan.

V. SARAN

Saran yang dapat digunakan untuk penelitian selanjutnya yakni pengujian tidak hanya untuk menguji fungsionalitas sistem saja, tetapi juga menguji performa, usabilitas, hingga keamanan sistem. Selain itu, dapat menggunakan *tools* lainnya dalam melakukan *automation testing*, misalnya Katalon Studio, Selenium, atau Robot Framework. Tak hanya itu, dalam menguji sistem juga dapat menggunakan teknik dalam *black box testing* lainnya, contohnya teknik *All Pair Testing*, *Fuzzing*, ataupun *Decision Table Testing*.

REFERENSI

- [1] R. K. Brahmana & R. Brahmana.(2013). *What Factors Drive Job Seekers Attitude in Using E-Recruitment*.The South East Asian Journal of Management, vol. 7(2), hal. 39-40.
- [2] L. Fajarwati, R. Hidayat, & Y. Budiarti. (2020). *Sistem Informasi Karyawan Baru PT Karya Putra Sapta Manunggal Berbasis Web*. Jurnal Teknologi Sistem Informasi dan Aplikasi, vol. 3(4), hal. 237-245.
- [3] W. Winata, A. W. R. Emanuel, & Herlina. (2022). *Pengujian Website EPOS PT XYZ Menggunakan Metode Black Box Testing*. Jurnal Informatika Atma Jogja, vol. 3(2), hal. 99-106.
- [4] Y. D. Wijaya & M. W. Astuti (2021). *Pengujian Black Box Sistem Informasi Penilaian Kinerja Karyawan PT INKA (PERSERO) Berbasis Equivalence Partitions*. Jurnal Digital Teknologi Informasi, vol. 4(1), hal. 22-26.
- [5] N. Mittal & N. Khandar. (2023). *Automation Testing Using Cucumber and Selenium Webdriver*. A Journal For New Zealand Herpetology, vol. 12(1), hal. 279-287.
- [6] M. Sholeh, I. Gifas, & M. A. Fauzi. (2021). *Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods*. Journal of Physics: Conference Series, vol. 12(29), hal. 1-8.
- [7] B. D. Saputra & A. Stefanie. (2023). *Automation Testing Api, Android, dan Website Menggunakan Serenity Bdd Pada Software Sistem Manajemen Rumah Sakit*. Jurnal Ilmiah Wahana Pendidikan, vol. 9(10), hal. 114-126.
- [8] M. Cernat, A. N. Staicu, & A. Stefanescu. (2020). *Improving UI Test Automation Testing using Robotic Process Automation*. Proceedings of the 15th International Conference on Software Technologies (ICSOF 2020), hal. 260-267.
- [9] A. Zulianto, A. Purbasari, N. Suryani, A. I. Susanti, F. R. Rinawan, & W. G. Purnama. (2021). *Pemanfaatn Katalon Studio untuk Otomatisasi Pengujian Black-Box pada Aplikasi iPosyandu*. JEPIN (Jurnal Edukasi dan Penelitian Informatika), vol. 7(3), hal. 370-378.
- [10] U. Nugraha, A. A. N. Robaiah, & D. Rospinoedji. (2020). *Testing The Information System Software Using Behavior Driven Development Method*. PalArch's Journal of Archaeology of Egypt/Egyptology, vol. 179(10), hal 3732-3742.