

Pemanfaatan *Boundary Value Analysis* dan *Equivalence Partitioning* pada *Automated Testing* Aplikasi Berbasis Website

Ery Setiyawan Jullev Atmadji^{1,*}, Ilham Robby Sanjaya², Hermawan Arief Putranto³
^{1,2,3}Department of Information Technology, State Politechnique Of Jember, Indonesia (9 pt)

Article Info

Article history:

Received May 15, 2023
Accepted May 30, 2023
Published May 30, 2023

Keywords:

Pengujian perangkat lunak
Pengujian Otomatis
Analisis nilai batas
Partisi kesetaraan
Skenario Kasus Uji

ABSTRAK

Pengujian perangkat lunak di bidang rekayasa perangkat lunak adalah proses siklus hidup dalam proyek perangkat lunak yang memverifikasi bahwa produk atau layanan memenuhi harapan kualitas dan memvalidasi bahwa perangkat lunak memenuhi spesifikasi dan persyaratan. Pengujian perangkat lunak dimaksudkan untuk menemukan cacat pada suatu program. Semakin kompleksnya sistem perangkat lunak saat ini mengakibatkan meningkatnya kebutuhan akan teknik pengujian yang canggih. Melakukan aktivitas pengujian perangkat lunak secara manual tampaknya tidak efektif dalam hal konsumsi tenaga kerja, yang memerlukan kecepatan eksekusi yang rendah dan cakupan tes yang tidak memadai. Untuk mengatasi permasalahan tersebut diperlukan pendekatan pengujian otomatis dengan salah satu metode yang dapat digunakan yaitu analisis nilai batas yang dipadukan dengan partisi ekuivalensi, sehingga dapat membantu proses pembuatan skenario kasus uji. Inputnya kurang valid sehingga aplikasi yang dibuat masih perlu perbaikan.



Corresponding Author:

Ery Setiyawan Jullev Atmadji,
Department of Information Technology,
State Politechnique Of Jember,
Jl. Mastrip PO BOX 164, Jember - Jawa Timur- Indonesia.
Email: *ery@polije.ac.id

1. PENGANTAR

Perangkat lunak memainkan peran yang semakin penting dalam sistem yang kompleks, terutama untuk aplikasi berteknologi tinggi yang terlibat dalam hal-hal penting pada berbagai bidang, seperti transportasi, bidang manajemen keuangan, komunikasi, aplikasi biomedis dan sebagainya. Pengujian perangkat lunak di bidang Rekayasa Perangkat Lunak adalah proses siklus hidup dalam proyek perangkat lunak yang memverifikasi bahwa produk atau layanan memenuhi harapan kualitas dan memvalidasi bahwa perangkat lunak memenuhi spesifikasi dan persyaratan[1]. Pengujian perangkat lunak dimaksudkan untuk menemukan cacat dalam suatu program kompleksitas yang berkembang dari sistem perangkat lunak saat ini menghasilkan peningkatan kebutuhan akan teknik pengujian yang canggih [2].

Pengujian perangkat lunak adalah Proses mengevaluasi perangkat lunak yang digunakan untuk mengidentifikasi kebenaran, kelengkapan dan kualitas dari perangkat lunak komputer yang dikembangkan untuk memeriksa apakah itu beroperasi seperti yang diharapkan untuk meningkatkan kualitas dan keandalannya. Pengujian Perangkat Lunak penting karena dapat menyebabkan kegagalan misi, berdampak pada kinerja operasional dan keandalan jika tidak dilakukan dengan benar. Pengujian perangkat lunak yang efektif memberikan perangkat lunak yang berkualitas produk yang memuaskan kebutuhan, kebutuhan, dan harapan pengguna [1].

Melakukan aktivitas pengujian perangkat lunak secara manual tampaknya tidak efektif dalam hal konsumsi tenaga kerja yang menuntut kecepatan eksekusi yang rendah, dan cakupan pengujian yang tidak memadai. Itulah masalah yang dapat diatasi oleh otomatisasi pengujian dan dapat dipecahkan juga dalam banyak kasus [3]. Kecerdasan buatan (AI) telah membuat dampak yang cukup besar pada bidang rekayasa perangkat lunak, dan bidang pengujian perangkat lunak tidak terkecuali. Secara teori, teknik AI dapat

membantu mencapai tingkat otomatisasi pengujian perangkat lunak setinggi mungkin [4]. Kegiatan ini dilaporkan telah ditingkatkan dengan menerapkan teknik AI sebagian besar dari pembelajaran mesin dan bidang visi komputer. Menurut studi utama yang ditinjau, peningkatan dicapai dalam hal penggunaan kembali kasus uji, pengurangan upaya manual, peningkatan cakupan, peningkatan deteksi kesalahan dan kerentanan. Beberapa alat yang disempurnakan dengan AI yang dapat diakses publik untuk otomatisasi pengujian perangkat lunak juga ditemukan selama peninjauan [2].

Aplikasi yang sudah diprogram sebegitu rupa harus menempuh tahap proses untuk memastikan tingkat kualitas dari perangkat lunak itu sendiri. Contoh proses yang dapat dikatakan baik kalau memiliki kesempatan mendapatkan sebuah error yang tidak dapat diketahui. Dari beberapa jenis metode pengujian Black Box salah satunya adalah Boundary Value Analysis [5]. Sedangkan Equivalence Partitions dapat membantu proses pembuatan case pengujian, uji kualitas dan menemukan kesalahan yang tidak terdeteksi yang disebabkan oleh kesalahan pengetikan [6]. Dalam penelitian ini penulis mengajukan metode blackbox testing dengan teknik boundary value analysis dan equivalence partitioning dalam menyelesaikan permasalahan tersebut. Adapun alasan dipilihnya kedua teknik tersebut untuk penelitian ini adalah karena keduanya relatif sederhana dibanding dengan tiga metode lainnya (decision table testing, state transition testing, dan use case testing) sebagai awalan. Banyak penelitian sebelumnya yang membahas tentang pengujian perangkat lunak secara otomatis dengan menggunakan bahasa Inggris seperti yang dilakukan [7] yang berjudul "Analysis of Artificial Intelligence Applications for Automated Testing of Video Games", [8] berjudul "The Journal of Systems and Software Automated code-based test selection for software product line regression testing".

Sehingga pada penelitian ini mengambil topik mengenai software testing dengan pendekatan *Boundary Value Analysis*, Teknik boundary value analysis bekerja dengan cara menentukan nilai batas atas dan batas bawah dari sebuah data yang akan diuji [9], selain dengan BVA model pengujian yang digunakan adalah *equivalence partitioning*, EP akan digunakan untuk mengetahui apakah Aplikasi masih dapat berfungsi dengan baik ketika menerima inputan berupa tipe data yang tidak diharapkan. Pengujian/ pada *partitioning equivalence* umumnya menggunakan test case yang telah ditentukan sebelumnya.

2. METODE PENELITIAN

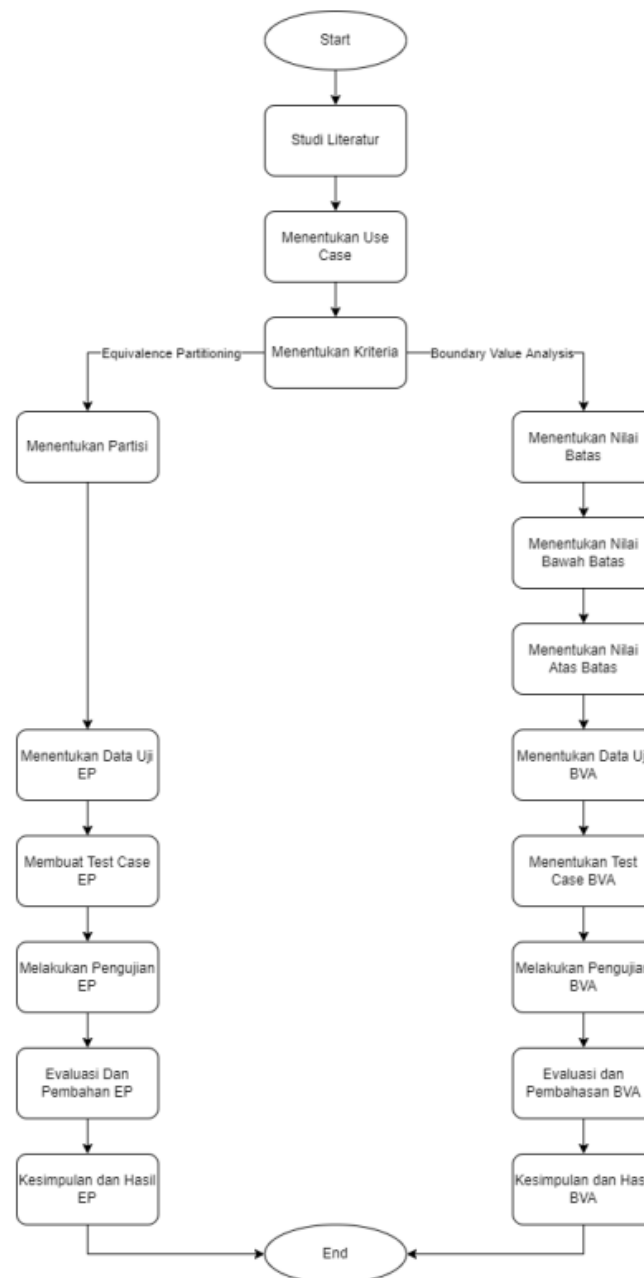
Pengujian perangkat lunak merupakan sebuah proses pengujian program yang dimaksudkan untuk mencari kesalahan pada software. Pengujian juga bertujuan untuk memastikan bahwa software memiliki kualitas yang baik. Kualitas software yang baik adalah software memenuhi kriteria yang diinginkan dan memberikan produktivitas yang tinggi, Pengujian perangkat lunak memiliki peranan penting dalam suatu sistem informasi, dengan pengujian ini dapat diketahui galat atau error yang akan muncul pada perangkat lunak. Dengan pengujian perangkat lunak diharapkan dapat meminimalisir kesalahan dan cacat pada sebuah software dan sebagai pengukuran kualitas dari software tersebut. Sehingga sangat perlu melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut

Selenium adalah tools auto testing yang digunakan untuk mengotomatisasi tes aplikasi web yang dilakukan pada browser. Selenium akan melakukan validasi web apps pada berbagai browser dan platform. Tools auto testing ini hanya dirancang untuk otomatisasi testing pada Web Apps, Jadi selain penggunaan untuk testing web apps, Misalnya menguji aplikasi desktop (perangkat lunak) atau menguji aplikasi seluler tidak akan bisa digunakan. Untuk membuat scriptnya kita bisa menggunakan beberapa bahasa pemrograman seperti Java, C #, Python dll. dikarenakan selenium bersifat open source kita bisa mengunduh dan menggunakan berbagai fitur yang tersedia pada selenium secara gratis. Selenium mendukung dan terintegrasi dari berbagai bahasa pemrograman, framework, dan berbagai platform pengembangan. Maka dari itu saat membuat script atau menggunakan selenium kita bebas menentukan akan menggunakan bahasa pemrograman yang mana. Salah satu teknik dari Black Box Testing yang akan digunakan untuk penelitian ini adalah Boundary Value Analysis (BVA). BVA akan digunakan untuk mengetahui apakah SAS-KIA masih dapat berfungsi dengan baik ketika menerima inputan berupa panjang data yang tidak diharapkan. Algoritma black box testing dengan teknik boundary value analysis adalah sebagai berikut :

1. Jika kondisi masukan berada pada kisaran nilai x , maka kasus uji harus dibuat dengan data sampel $x-1$, x , dan $x+1$.
2. Jika kondisi masukan yang menggunakan sejumlah nilai, maka kasus uji harus dibuat data sampel minimum -1 , minimum, maksimum, maksimum $+1$.
3. Lakukan langkah 1 dan 2 untuk proses output.
4. Jika data telah memiliki batasan masukan (misal: larik ditetapkan maks. 10), maka kasus uji dibuat pada batasan tersebut.

Salah satu teknik dari Black Box Testing yang akan digunakan untuk penelitian ini adalah Equivalence Partitioning (EP). EP akan digunakan untuk mengetahui apakah SAS-KIA masih dapat berfungsi dengan baik ketika menerima inputan berupa tipe data yang tidak diharapkan. Pengujian/ pada *partitioning equivalence* umumnya menggunakan test case. Sama seperti BVA, hal penting dalam test case

partitioning equivalence adalah conclusion yang berisi success atau failed-nya pengujian tersebut. Pada penelitian tersebut terdapat flowchart yang akan menjelaskan mengenai beberapa pendekatan yang akan digunakan pada penelitian kali ini.



Gambar 1 Flowchart Penelitian

Berdasarkan flow yang telah di buat tersebut, masing-masing pengujian membutuhkan test scenario yang juga berbeda-beda, oleh karena itu maka pada masing-masing test scenario akan dilakukan pengujian dengan jumlah uji yang sama antara satu scenario dengan scenario yang lain sehingga jumlah pengujian akan sama namun hasil dari pengujian akan berbeda.

3. HASIL DAN ANALISIS

Pada beberapa penelitian terdahulu yang juga berfokus pada software defect testing menggunakan pendekatan yang sangat beragam, salah satunya adalah dengan menggunakan naïve bayes pada proses pengujian software [7], pada penelitian tersebut Naïve bayer mampu melakukan prediksi software dengan tingkat akurasi 75%, sedangkan pada penelitian lain menggunakan pendekatan BVA pada aplikasi voucher dan receipt namun dengan jumlah test scenario sebanyak 57 scenario dihasilkan akurasi sebesar 75.95% [10].

Dengan dua penelitian tersebut maka pada penelitian kali ini BVA tidak hanya berdiri sendiri namun juga dilakukan kombinasi dengan *Equivalence Partitioning* yang memiliki kelebihan pada saat pembuatan test scenario yakni dengan membagi data pengujian menjadi beberapa bagian input yang model inputannya tidak sesuai dengan apa yang diharapkan guna mengetahui keefektifan serta kehandalan sistem dalam menangani error yang terjadi [11].

Test case untuk pengujian BVA, contoh pada *form login* pada *field username* dan *password* dibagi ke dalam 6 buah kondisi, yaitu tipe inputan adalah nilai batas bawah, nilai batas, dan nilai batas atas. Seperti dapat dilihat pada Gambar 2 untuk password. Keenam tipe data ini diambil dari data uji form login yang telah dijelaskan pada bagian lainnya. Baik pada pengujian dengan tipe inputan nilai batas bawah, nilai batas, maupun nilai batas atas, *system* diberikan waktu dengan memberikan perintah *implicitly wait* untuk menunggu *object* yang akan diinputkan terlihat. Hal ini dibutuhkan agar pengujian dapat berjalan dengan lebih efisien karena seringkali, terdapat kendala pada saat *system* memuat halaman yang dituju. Hal ini dapat disebabkan oleh beberapa hal tertentu, salah satunya adalah lambatnya koneksi internet. Setelah *object* untuk menerima inputan ditemukan, *system* memberikan inputan baik dengan nilai batas bawah, nilai batas, maupun nilai batas atas. Umumnya, *system* tidak akan memberikan feedback ataupun warning apapun jika inputan yang dimasukkan sesuai dengan nilai batas. Namun sebaliknya, jika inputan yang dimasukkan sudah melewati karakter maksimal atau disebut dengan nilai batas atas dan nilai batas bawah maka *system* akan memberikan warning berupa 'Jumlah karakter melebihi batas' untuk batas atas dan 'Jumlah karakter terlalu sedikit' untuk batas bawah. Untuk itu pada kondisi tipe inputan nilai batas atas terdapat tahap dimana akan diverifikasi apakah *system* dapat menunjukkan bahwa terdapat atribut yang memberikan warning.

```
def test_username_password_valid(self):
    driver = self.driver
    driver.get("https://sas-kia-deploy.vercel.app/login")
    driver.find_element_by_id("email").click()
    driver.find_element_by_id("email").clear()
    driver.find_element_by_id("email").send_keys("agoes.soetanto2@gmail.com")
    driver.find_element_by_id("password").click()
    driver.find_element_by_id("password").clear()
    driver.find_element_by_id("password").send_keys("5@AG0ES67s")
    driver.find_element_by_id("mui-1").click()
    self.assertEqual("Login Sukses!", self.close_alert_and_get_its_text())
    driver.get("https://sas-kia-deploy.vercel.app/landing-page")
    driver.find_element_by_xpath("//div[@id='root']/div/header/div/div[2]/ul/li[5]/a/span").click()
```

Gambar 2 test Case Scenario dengan pendekatan BVA

Test case untuk pengujian EP, seperti dicontohkan pada form login pada field username dan password dibagi ke dalam 2 buah kondisi, yaitu ketika tipe inputan adalah valid dan invalid yang dapat dilihat pada gambar 3. Kedua tipe data ini diambil dari data uji form login yang telah dijelaskan pada sub bab sebelumnya. Baik pada pengujian dengan tipe inputan valid maupun invalid, *system* diberikan waktu dengan memberikan perintah *implicitly wait* untuk menunggu *object* yang akan diinputkan terlihat. Hal ini dibutuhkan agar pengujian dapat berjalan dengan lebih efisien karena seringkali, terdapat kendala pada saat *system* memuat halaman yang dituju. Hal ini dapat disebabkan oleh beberapa hal tertentu, salah satunya adalah lambatnya koneksi internet. Setelah *object* untuk menerima inputan ditemukan, *system* memberikan inputan baik dengan inputan valid maupun invalid. Setelah berhasil memberikan inputan, maka *system* akan diminta untuk menekan button masuk. Jika *system* menerima inputan invalid maka *system* akan diminta untuk memverifikasi apakah terdapat element yang memberikan warning bahwa username / password tidak boleh kosong. Sama halnya dengan inputan valid, *system* akan diminta menunggu apakah halaman saat inputan valid benar-benar sudah muncul atau belum. Jika sudah, maka *system* akan diminta memverifikasi apakah element tersebut benar-benar terlihat pada antarmuka website yang telah diuji.

```
def test_password_invalid_wrong(self):
    driver = self.driver
    driver.get("https://sas-kia-deploy.vercel.app/login")
    driver.find_element_by_id("email").click()
    driver.find_element_by_id("email").clear()
    driver.find_element_by_id("email").send_keys("agoes.soetanto2@gmail.com")
    driver.find_element_by_id("password").click()
    driver.find_element_by_id("password").clear()
    driver.find_element_by_id("password").send_keys("5@AG0ES67")
    driver.find_element_by_id("mui-1").click()
    self.assertEqual("Login Gagal Silakan Periksa Kembali Username dan Password Anda!", self.close_alert_and_get_its_text())
```

Gambar 3 Test Case Scenario dengan pendekatan EP

Pada form login, terdapat 3 kali pengujian BVA yang masing-masing dilakukan dengan memasukkan nilai batas, nilai bawah batas, dan nilai atas batas. Pada pengujian yang dilakukan dengan memasukkan nilai batas pada field username dan password, pengujian memberikan hasil dimana test case yang dijalankan berhasil / pass yang dapat dilihat pada gambar 4a dan 4b

Command	Target	Value
open	https://saa-kia-deploy.vercel.app/Login	
click	id=email	
type	id=email	agoes.soetanto2@gmail.com

```
[info] Executing: | click | id=email | |
[info] Executing: | type | id=email | agoes.soetanto2@gmail.com |
[info] Time: Sat Jul 23 2022 16:05:45 GMT+0700 (Western Indonesia Time) Timestamp: 1658567145543
[info] Test case passed
```

Command	Target	Value
open	https://saa-kia-deploy.vercel.app/Login	
click	id=password	
type	id=password	5gAG0E567s

```
[info] Executing: | click | id=password | |
[info] Executing: | type | id=password | 5gAG0E567s |
[info] Time: Sat Jul 23 2022 16:06:54 GMT+0700 (Western Indonesia Time) Timestamp: 1658567214467
[info] Test case passed
```

Gambar 4a Hasil pengujian dengan BVA Login Batas Bawah Gambar 4b Hasil Pengujian dengan BVA Login Batas Atas

Seperti yang telah dijelaskan pada poin-poin sebelumnya, form login mempunyai 2 buah field yang dapat menerima inputan, yaitu username dan password. Setelah dilakukan 6 kali pengujian, yaitu pengujian dengan memasukkan nilai batas, nilai batas bawah, dan nilai batas atas berdasarkan test case, field username memberikan hasil yang signifikan. Pada pengujian yang dilakukan dengan memasukkan nilai batas, test case berhasil / pass. Sesuai dengan yang tertulis dalam test case, bahwa ketika inputan tersebut termasuk pada nilai batas, aplikasi tidak memberikan respon tertentu. Hal ini dikarenakan inputan tersebut memang sudah sesuai sehingga tidak ada feedback apapun dari sistem. Kemudian, pada pengujian yang dilakukan dengan memasukkan nilai bawah batas, test case tersebut gagal / failed. Hal ini disebabkan karena Aplikasi tidak memberikan warning kepada user bahwa jumlah karakter yang diinputkan kurang dari limit. Terakhir, pada pengujian yang dilakukan dengan memasukkan nilai atas batas, test case tersebut gagal / failed. Hal ini disebabkan karena Aplikasi tidak memberikan warning kepada user bahwa jumlah karakter yang diinputkan sudah melebihi limit. Test case yang fail ini secara langsung juga menandakan bahwa masih terdapat bug pada field username dan password, khususnya pada saat memasukkan karakter yang melebihi jumlah inputan maximal, dimana seharusnya Aplikasi memberikan feedback berupa warning ketika inputan yang diberikan oleh user sudah melebihi batas maximal inputan. Hasil dari pengujian yang dilakukan dengan pendekatan BVA menghasilkan data seperti terlihat pada Tabel 1.

Tabel 1 Hasil Analisa Pengujian dengan BVA

Kode	Nama Field	Nilai Batas	Nilai Batas Bawah	Nilai Batas Atas
TC01	Username	passed	failed	failed
TC02	Password	passed	failed	failed
TC03	Nama Lengkap	passed	failed	failed
TC04	Email	passed	failed	failed
TC05	Alamat	passed	failed	failed
TC06	Nip	failed	failed	passed
TC07	Nomor Telepon	failed	failed	passed
TC08	Password	passed	failed	failed
TC09	Search Pengguna	passed	failed	failed
TC10	Search Kia	passed	failed	failed
TC11	Search Imunisasi	passed	failed	failed
TC12	Search Gizi	passed	failed	failed
TC13	Search Gizi Coc	passed	failed	failed
TC14	Search Coc Indikator Ibu	passed	failed	failed

Jumlah form yang diuji sebanyak 8 form, dengan field inputan sebanyak 14 form, dan hasil pengujian nilai batas menunjukkan sebanyak 12 field mendapatkan hasil passed dan 2 field mendapatkan hasil failed, kemudian dari semua field yang diuji pada nilai batas bawah semua mendapatkan hasil failed dikarenakan sistem tidak memunculkan warning ketika nilai batasnya kurang dari batas bawah yang ditentukan, dan pada nilai batas atas terdapat 2 field yang menunjukkan hasil passed dan 12 sisanya mendapatkan hasil failed.

Setelah dilakukan evaluasi kepada masing-masing field yang dapat menerima inputan dengan nilai batas maximal, dapat dilihat bahwa Aplikasi masih membutuhkan beberapa perbaikan dalam mengatasi user saat memasukkan jumlah inputan yang karakternya melebihi limit. Sesuai dengan apa yang didapat dari hasil pengujian, pada keenam form yang diuji, hanya 2 field yang bisa mengatasi karakter yang melebihi limit namun itu pun masih tidak menampilkan pesan warning dari sistem. Selebihnya, field tidak memberikan warning pada saat user memasukkan inputan yang jumlah karakternya melebihi limit. Oleh sebab itu, Aplikasi masih perlu memperbaiki sistemnya, dengan menambahkan warning ataupun notifikasi yang dapat

memberitahu user bahwa jumlah karakter yang diinputkannya telah melebihi limit. Hal ini tidak bisa dianggap sepele, karena seringkali terjadi kondisi dimana user memberikan sebuah inputan namun inputan tersebut tiba-tiba tidak bisa diinputkan ke dalam sebuah field akan tetapi user tidak tahu kenapa inputan nya tidak bisa diinputkan lagi sehingga menyebabkan berbagai asumsi dari pihak user.

Dengan memberikan warning bahwa inputannya telah melewati batas maksimal, maka user dapat mengetahui ketika inputan yang ia masukkan tidak bisa diinputkan lagi karena hal itu disebabkan oleh jumlah karakternya yang sudah melebihi limit ataupun alasan lainnya. Kemudian, sesuai yang telah dijabarkan pada poin-poin kriteria keenam form tersebut, masih banyak inputan yang batas maksimal nya adalah 250 karakter. Ada baiknya karakter ini juga disesuaikan dengan kebutuhan field yang dituju, seperti Username, nama lengkap, password dll. Username, nama lengkap, dan password pada Aplikasi sendiri memiliki rata-rata panjang karakter yang tidak lebih dari 30 namun batas maksimal inputan nya adalah 250. Batas tersebut terlalu jauh dari jumlah inputan yang sebenarnya sehingga akan lebih baik jika panjang inputan maksimal karakternya dibuat menjadi 30 pula. Kemudian yang tidak kalah penting pada form nip dan nomor telepon memang sudah terlihat bagus dalam hal membatasi inputan, namun batas tersebut terlalu sedikit terutama pada nomor telepon dimana batasnya hanya 11 nomor hal tersebut sangat mempengaruhi website, dikarenakan beberapa nomor telepon tidak selalu berjumlah 11 bisa jadi lebih dari itu, hal ini yang perlu diperhatikan betul untuk para developer dalam membangun sebuah website.

Berdasarkan kriteria yang telah dijabarkan pada sub bab sebelumnya, maka form login memiliki 2 buah field yang dapat menerima inputan. Kedua field tersebut adalah username dan password. Username sendiri memiliki nilai inputan valid, yaitu Email dengan panjang data maksimal 50, sedangkan inputan tak valid-nya adalah null dan wrong / salah. Disisi lain, field password memiliki inputan valid berupa text dengan panjang 50. Hal ini membuat password jug mempunyai inputan invalid berupa null dan wrong pula. Inputan-inputan valid dan invalid tersebut dibagi ke dalam 2 buah partisi, yaitu valid dan invalid partition yang dapat dilihat pada tabel 4.50

Tabel 2 Test Scenario pada pengujian EP

Kode	Nama Field	Valid Partition	Invalid Partition
P01	Username	[Email] max 50	Null,Wrong
P02	Password	[Text] max 50	Null,Wrong

Pada form login, terdapat 3 kali pengujian EP yang masing masing dilakukan dengan memasukkan nilai valid dan invalid. pada pengujian yang dilakukan dengan memasukkan nilai invalid pada field username dan password, pengujian memberikan hasil dimana test case yang dijalankan gagal / failed yang dapat dilihat pada gambar 5 untuk username dan password ini dikarenakan sistem menolak user untuk login dikarenakan data yang dimasukkan salah dan harus login kembali, sedangkan pada gambar 6 sukses menjalankan testing dengan skenario EP.

Command	Target	Value
open	https://saa-ksa-deploy.xenol.app/landing-page	
click	btn-Masuk	
open	https://saa-ksa-deploy.xenol.app/Login	
click	id=password	
type	id=password	5gAG0ES67s
click	id=mul-4	
assertAlert	Login Gagal Silakan Periksa Kembali Username dan Password Anda!	

```

[info] Executing: | click | id=password | |
[info] Executing: | type | id=password | 5gAG0ES67s |
[info] Executing: | click | id=mul-1 | |
[info] Executing: | assertAlert | Login Gagal Silakan Periksa Kembali Username dan Password Anda! |
|
[error] No response!!!!
[info] Time: Sat Jul 23 2022 17:11:17 GMT+0700 (Western Indonesia Time) Timestamp: 1658571077360
[info] Test case failed
[info] Stop executing

```

Gambar 5 error Executing in EP

Command	Target	Value
open	https://sas-kia-deploy.vercel.app/Login	
click	id=email	
type	id=email	egres.soearto2@gmail.com
click	id=password	
type	id=password	5@AGOE567
click	id=mui-1	
assertAlert	Login Gagal Silakan Periksa Kembali Username dan Password Anda!	

```

[info] Executing: | click | id=password | |
[info] Executing: | type | id=password | 5@AGOE567 |
[info] Executing: | click | id=mui-1 | |
[info] Executing: | assertAlert | Login Gagal Silakan Periksa Kembali Username dan Password Anda! |
|
[info] Time: Sat Jul 23 2022 17:17:05 GMT+0700 (Western Indonesia Time) Timestamp: 1658571425382
[info] Test case passed

```

Gambar 6 EP Testing scenario Success

Seperti yang telah dijelaskan pada poin-poin sebelumnya, form login mempunyai 2 buah field yang dapat menerima inputan, yaitu username dan password. Setelah dilakukan 4 kali pengujian, yaitu pengujian dengan memasukkan nilai valid dan invalid kepada 2 field tersebut, berdasarkan test case, field username dan password memberikan hasil yang signifikan. Pada pengujian yang dilakukan dengan memasukkan nilai valid maupun invalid, test case berhasil / pass dimana artinya test case tersebut berjalan dengan baik sesuai apa yang diharapkan. Hal ini tentunya sejalan dengan yang tertulis dalam test case, bahwa ketika sudah menerima inputan yang valid maupun sebaliknya, Aplikasi Web harus bisa menunjukkan bahwa terdapat element yang menunjukkan bahwa validasi tersebut memberikan feedback kepada user. Validasi ini dapat berupa notifikasi alert yang dapat memberi tahu user bahwa inputan yang telah diberikan sesuai maupun sebaliknya. Pada inputan invalid, dengan adanya notifikasi alert ini, sistem dapat dikatakan sudah bisa mencegah user ketika memasukkan inputan yang tidak sesuai kriterianya. Sebaliknya, notifikasi saat user memberikan inputan yang benar juga diperlukan agar user tahu dan tidak bingung ketika sistem tiba-tiba berpindah halaman. Dengan hasil test case yang berhasil ini, Aplikasi Web dapat dikatakan sudah dapat mencegah user dalam memberikan inputan yang tidak valid dan sebaliknya, pada hasil pengujian tersebut didapatkan hasil dari EP dapat dilihat pada Tabel 3.

Tabel 3 Hasil pengujian Aplikasi dengan pendekatan EP

Kode	Invalid Partition	Hasil Yang Diharapkan	Hasil Sebenarnya	Hasil
TC01	Username Null	Failed	Failed	Passed
TC02	Username Salah	Failed	Failed	Passed
TC03	Password Null	Failed	Failed	Passed
TC04	Password Salah	Failed	Failed	Passed
TC05	Nama Lengkap Special Character	Failed	Failed	Passed
TC06	Nama Lengkap Null	Failed	Failed	Passed
TC07	Email Tanpa @	Failed	Failed	Passed
TC08	Email Tanpa .com	Failed	Failed	Passed
TC09	Email Null	Failed	Failed	Passed
TC10	Alamat 1 Character	Failed	Passed	Failed
TC11	Alamat Null	Failed	Failed	Passed
TC12	Nip Special Character	Failed	Failed	Passed
TC13	Nip Null	Failed	Failed	Passed
TC14	No Telepon Special Character	Failed	Failed	Passed
TC15	No Telepon Null	Failed	Failed	Passed
TC16	Password 5 Character	Failed	Failed	Passed
TC17	Password Null	Failed	Failed	Passed
TC18	Search Pengguna 1 Character	Failed	Passed	Failed
TC19	Search Pengguna Null	Failed	Failed	Passed
TC20	Search Kia Special Character	Failed	Passed	Failed
TC21	Search Imunisasi Special Character	Failed	Passed	Failed

Tentunya, setelah dilakukan evaluasi kepada masing-masing field yang dapat menerima inputan pada equivalence partitioning, dapat dilihat bahwa Aplikasi Website yang dibangun masih membutuhkan perbaikan dalam mengatasi user saat memasukkan tipe inputan yang tidak sesuai, yaitu pada field search data pengguna, search data kia, search data imunisasi, search data gizi, search data gizi coc, search data coc indikator ibu. Oleh sebab itu, Aplikasi Website tersebut masih perlu memperbaiki beberapa bug yang telah ditemukan, dengan menambahkan *warning* ataupun notifikasi yang dapat memberitahu user bahwa tipe inputan nya tidak sesuai. Dengan memberikan *warning* bahwa tipe inputannya tidak sesuai, maka user dapat mengetahui ketika inputan yang dimasukkan tidak bisa diproses. Selain itu, Aplikasi Website yang dibangun akan mampu meminimalisir kesalahan inputan data. Kemudian, sesuai dengan apa yang didapat dari hasil pengujian, pada kedelapan form yang diuji, semuanya memiliki field yang tidak memberikan *warning* pada saat user memasukkan inputan yang tipe inputannya valid. Terakhir, pada aplikasi yang dibuat juga dapat meng-enhance *wording* pada notifikasi-notifikasi yang di tampilkan. Seringkali sistem sudah benar, yaitu memberikan *warning* ketika inputan yang dimasukkan tidak sesuai namun pesannya yang disampaikan tidak

jelas karena pesannya terlalu general dan tidak spesifik seperti pada form tambah pengguna dimana setelah menginputkan data invalid muncul 2 alert yang memberitahukan kalau data berhasil ditambahkan dan muncul lagi data gagal ditambahkan, hal ini yang perlu segera ditindak lanjuti karena membuat user bingung akan data inputannya sudah berhasil ditambahkan atau ketika gagal dalam melakukan input data.

4. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilaksanakan dan pengujian yang telah dilakukan pada aplikasi yang telah dikembangkan, mendapatkan kesimpulan bahwasanya aplikasi yang telah dikembangkan mempunyai permasalahan yakni belum memberikan warning ketika menerima inputan dengan panjang data yang tidak valid, khususnya pada saat menerima inputan yang melewati nilai batas maximal dan minimal karakter yang dapat diinputkan, selain itu Website yang telah dikembangkan belum sepenuhnya memberikan warning pada saat menerima inputan dengan tipe data invalid, hal ini dapat dibuktikan dimana masih ada 7 field yang tidak memberikan warning saat inputannya invalid sehingga akan memudahkan dalam melakukan proses pengecekan input pengguna. Selain itu dengan menerapkan automated testing menggunakan teknik *boundary value analysis* yang disertai dengan *equivalence partitioning* untuk menguji website lebih mudah, cepat dan tepat. Saran yang bisa dilakukan pada proses selanjutnya adalah mengupayakan agar pengujian dapat dilakukan secara otomatis dengan menggunakan pendekatan yang lain seperti *Machine Learning* Maupun dengan pendekatan *Multi agent system*, sehingga diharapkan pengujian akan lebih mudah dan meminimalisir kesalahan pada saat pembuatan testing scenario pada sebuah test scenario.

DAFTAR PUSTAKA

- [1] E. Sianturi, "Boundary Value Analysis and Decision Table Testing Methods in Software Testing Enjelina Sianturi," *Int. J. Inf. Technol. Educ.*, vol. 1, no. 3, pp. 2809–8463, 2022.
- [2] A. Firmansyah, M. A. Arief, M. Daffa, F. Falah, O. D. Dharmawan, and J. Riyanto, "Pengujian Aplikasi Sistem Penilaian Mahasiswa Dengan Menggunakan Teknik Boundary Value Analysis," 2022. [Online]. Available: <http://pijarpemikiran.com/index.php/Scientia>.
- [3] A. Rifqi, Y. Arfani, P. Kasih, and D. P. Pamungkas, "Pengujian Aplikasi Presensi dengan Black box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis."
- [4] C. Aritama and A. Priadana, "IMPLEMENTATION AND PERFORMANCE ANALYSIS OF PRIVATE CLOUD USING OPENSTACK SWIFT DAN RCLONE," vol. III, no. IX, pp. 317–322, 2018, doi: 10.9790/0661-1805064858.
- [5] I. D. M. Widia, S. Rosalin, S. R. Asriningtias, and E. Sonalita, "BlackBox Testing Menggunakan Boundary Value Analysis dan Equivalence Partitioning pada Aplikasi Pengadaan Bahan Baku Batik dengan Pendekatan Use Case," *JIMP J. Inform. Merdeka Pasuruan*, vol. 7, no. 2, pp. 15–21, 2022.
- [6] Y. Ike Melani, "Black Box Testing Using Equivalence Partition Method in Sintana Application," 2021.
- [7] I. Zarembo, "Analysis of artificial intelligence applications for automated testing of video games," *Vide. Tehnol. Resur. - Environ. Technol. Resour.*, vol. 2, pp. 170–174, 2019, doi: 10.17770/etr2019vol2.4158.
- [8] P. Jung, S. Kang, and J. Lee, "Automated code-based test selection for software product line regression testing," *J. Syst. Softw.*, vol. 158, 2019, doi: 10.1016/j.jss.2019.110419.
- [9] F. D. Hartono and Y. Sugiarti, "PERBANDINGAN METODE EQUIVALENCE PARTITIONS DAN BOUNDARY VALUE ANALYSIS PADA PENGUJIAN BLACK BOX Literatur Review," *Maj. Ilm. METHODA*, vol. 12, no. 2, pp. 153–159, Aug. 2022, doi: 10.46880/methoda.Vol12No2.pp153-159.
- [10] Y. Cahyaningrum and I. R. Widiyari, "Analisis Performa Container Berplatform Docker atas Serangan Malicious Software (Malware)," *J. Buana Inform.*, vol. 11, no. 1, p. 47, 2020, doi: 10.24002/jbi.v11i1.3279.
- [11] M. Sholeh, I. Gisfas, Cahiman, and M. A. Fauzi, "Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods," in *Journal of Physics: Conference Series*, Mar. 2021, vol. 1823, no. 1, doi: 10.1088/1742-6596/1823/1/012029.