



AUTOMATION TEST MENGGUNAKAN TEKNIK BEHAVIOR DRIVEN DEVELOPMENT PADA APLIKASI GA SYSTEM 2 UNITED TRACTORS

Niken Arra Ilma Auliya¹⁾, Addin Aditya^{2)*}, Diah Arifah P.³⁾

^{1,3}Program Studi Informatika, Sekolah Tinggi Informatika & Komputer Indonesia

²Program Studi Sistem Informasi, Sekolah Tinggi Informatika & Komputer Indonesia

^{1,2,3}Jl. Raya Tidar No 100, Kota Malang 65146

Email: ¹191116014@mhs.stiki.ac.id, ^{2*}addin@stiki.ac.id, ³diah@stiki.ac.id

Abstract

Software development that is complex and prone to failures, such as bugs or defects, demands effective testing to ensure quality and reduce the risk of failure during release. This research uses the automation testing method with behavior-driven development (BDD) techniques on the United Tractors GA System 2 Application, focusing on the tender feature. This research aims to test the role of BDD techniques in implementing automation testing on the United Tractors GA System 2 Application and compare test results between automation testing and manual testing methods. The test results show that implementing automation testing with BDD techniques on the United Tractors GA System 2 Application can speed up the testing process, increase time efficiency in application development, and reduce the risk of bugs when released to users. Automation testing also shows a better time ratio, with a ratio of 4:15 on the United Tractors GA System 2 Application tender feature. Based on the research results, it can be concluded that using automation testing with BDD techniques is an effective and efficient approach in software testing, especially in testing the tender feature in the United Tractors GA System 2 Application. The research objective of finding better solutions in software testing has been achieved. Implementing automated testing using BDD techniques on the United Tractors GA System 2 Application provides significant benefits in speeding up the testing process, increasing the efficiency of application development time, and reducing the risk of bugs during release to users. This research proves that testing automation with a BDD approach is an effective solution for improving the quality and overall quality of software testing.

Keyword: Automation Test, Manual Test, Behavior Driven Development, Katalon Studio

Abstrak

Pengembangan perangkat lunak yang kompleks dan rentan terhadap kegagalan, seperti bug atau cacat, menuntut pengujian yang efektif guna memastikan kualitas dan mengurangi risiko kegagalan saat rilis. Proses pengujian perangkat lunak pada aplikasi GA system 2 United Tractors saat ini masih menggunakan pengujian manual sehingga masih tergantung dengan peran manusia dan hal ini berdampak pada sisi keakuratan hasil pengujian. Penelitian ini bertujuan untuk menguji peran teknik BDD dalam implementasi automation testing pada Aplikasi GA System 2 United Tractors serta membandingkan hasil pengujian antara metode automation testing dan manual testing. Penelitian ini menggunakan metode automation testing dengan teknik behavior driven development (BDD) pada Aplikasi GA System 2 United Tractors, dengan fokus pada fitur tender. Hasil pengujian menunjukkan bahwa implementasi automation testing dengan teknik BDD pada Aplikasi GA System 2 United Tractors dapat mempercepat proses pengujian, meningkatkan efisiensi waktu dalam pengembangan aplikasi, dan mengurangi risiko bug saat rilis kepada pengguna. Pengujian automation juga menunjukkan perbandingan waktu yang lebih baik, dengan rasio 4:15 pada fitur tender Aplikasi GA System 2 United Tractors. Berdasarkan hasil penelitian, dapat disimpulkan bahwa penggunaan automation testing dengan teknik BDD merupakan pendekatan yang efektif dan efisien dalam pengujian perangkat lunak, terutama dalam menguji fitur tender pada Aplikasi GA System 2 United Tractors. Penelitian ini membuktikan bahwa pengujian automation dengan pendekatan BDD merupakan solusi yang efektif dalam meningkatkan kualitas dan keseluruhan pengujian perangkat lunak.

Kata Kunci: Automation Test, Manual Test, Behavior Driven Development, Katalon Studio

1. PENDAHULUAN

Dalam proses pengembangan perangkat lunak, prosesnya cenderung kompleks dan berpotensi mengalami sebuah kegagalan, termasuk kemungkinan munculnya bug atau cacat dalam perangkat lunak. Secara umum proses pengembangan perangkat lunak saat ini lebih cenderung memilih pendekatan seperti agile atau scrum. Adapun tahap pengembangan meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, integrasi, implementasi,



pemeliharaan, dan peningkatan. Pada tahap proses pengembangan aplikasi tersebut terdapat tahap yang sangat penting yaitu tahap pengujian atau testing. Testing mencakup aktivitas siklus hidup perangkat lunak yang terkait dengan perencanaan, persiapan, dan evaluasi untuk memastikan bahwa perangkat lunak memenuhi persyaratan dan dapat mendeteksi serta mengenali cacat atau bug [1] [2] [3]. Pengujian perangkat lunak dapat dilakukan secara manual maupun automation. Aplikasi GA System 2 United Tractors adalah perangkat lunak berbasis website yang membantu pengguna dalam permintaan kebutuhan general affair dan project management pada proyek konstruksi. Pada Aplikasi GA System 2 United Tractors terdapat salah satu fitur penting yaitu fitur tender. Dimana fitur ini memiliki kepentingan yang tinggi dan perlu dilakukan pengujian yang komprehensif untuk memastikan kualitas fitur tersebut. Kondisi saat ini proses pengujian aplikasi tersebut hanya melakukan pengujian secara manual testing dan memprioritaskan waktu lebih untuk melakukan pengujian. Pengujian secara manual testing ini dirasa tidak efektif dan efisien untuk dilakukan ketika dalam tahap regression testing atau pengujian berulang, sehingga pengujian tidak maksimal untuk menghindari bug atau cacat ketika rilis ke pengguna. Selain itu pelaporan report bug juga dilakukan secara manual melalui dokumentasi excel. Dalam hal tersebut sangat berdampak pada kurangnya kualitas produk pada Aplikasi GA System 2 United Tractors dan kolaborasi tim akan mengalami keterlambatan dalam melakukan monitoring perbaikan produk.

Menurut penelitian oleh Kusyadi et al. (2022) berjudul "Pengujian Aplikasi Kas Keuangan Menggunakan Katalon", pengujian aplikasi kas keuangan dilakukan menggunakan Katalon Studio. Tujuan penelitian ini adalah menguji kualitas website dan aplikasi kas keuangan agar error-free dengan metode pengujian GUI, termasuk backend performance test, frontend performance test, functional test, dan user interface test. Penelitian ini menyimpulkan bahwa Katalon Studio sangat membantu dalam menguji dan meningkatkan fitur-fitur utilitas pengujian website, dengan pengujian GUI menjadi faktor penilaian kualitas [3]. Selanjutnya, penelitian oleh Zualianto et al. (2021) berjudul "Pemanfaatan Katalon Studio Untuk Otomatisasi Pengujian Black-Box Pada Aplikasi Iposyandu" meneliti pengujian otomatis pada aplikasi Iposyandu. Tujuan penelitian ini adalah melaporkan hasil pengujian otomatis menggunakan Katalon Studio pada aplikasi iPosyandu. Penelitian ini menunjukkan bahwa pengujian otomatis dengan Katalon Studio meningkatkan efektivitas pengujian dan mengurangi kesalahan manusia. Selain itu, penelitian ini juga menyimpulkan adanya fungsionalitas yang perlu diperbaiki dan kebutuhan untuk menambahkan fungsionalitas lain [4]. Penelitian oleh (Kosasih & Cahyono, 2021) berjudul "Automation testing tool dalam pengujian aplikasi the point of sale" meneliti pengujian otomatis pada aplikasi The Point of Sale (TPOS). Tujuannya adalah menguji kualitas perangkat lunak berbasis web menggunakan metode pengujian Blackbox pada aplikasi TPOS. Penelitian ini mencakup pengujian kualitas perangkat lunak melalui uji fungsional, kinerja antarmuka depan dan belakang, serta user interface test. Penelitian ini menyimpulkan bahwa pengujian otomatis dengan skrip otomatisasi diperlukan dalam pengujian yang berulang-ulang dan melibatkan berbagai platform dan data yang besar [5]. Penelitian oleh (Rusli, 2020) berjudul "Analisa Perbandingan Black-box Automated Testing dan Manual Testing pada Aplikasi ACCMART" meneliti pengujian bug pada aplikasi ACCMart dan membandingkan pengujian manual dengan pengujian otomatis. Penelitian ini terbatas pada aplikasi ACCMart dan hanya menguji sejumlah fungsi yang telah dipilih sebelumnya. Pengujian dilakukan menggunakan Katalon Studio dengan metode black-box. Hasil penelitian menunjukkan perbedaan waktu dan faktor penting antara pengujian otomatis dan manual [6].

Berdasarkan penjabaran permasalahan serta beberapa kajian ilmiah di atas, penelitian ini bertujuan untuk melakukan pengujian menggunakan automation testing dengan tools katalon studio dengan teknik Behavior Driven Development (BDD) pada aplikasi GA System 2 United Tractors. Pemilihan katalon studio sebagai tools pengujian automation didasarkan pada performa yang baik, termasuk kemampuan pengujian secara cross-browser, menjadi all in one test solution dan end-to-end automation platform. Dalam pengembangan perangkat lunak, penambahan fitur baru dan kerjasama stakeholder menjadi penting. Salah satu teknik yang cocok untuk solusi pengujian automation adalah Behavior Driven Development (BDD), yang merupakan metode pengembangan perangkat lunak dalam pendekatan agile. BDD mengedepankan prinsip "test-first" dengan mengembangkan kasus uji berdasarkan kebutuhan stakeholder [7]. Dengan menerapkan teknik BDD, diharapkan implementasi pengujian perangkat lunak dapat dipahami oleh pelaku bisnis dan orang teknis karena didasarkan pada alur bahasa yang digunakan oleh pengguna. Dengan demikian, tujuan penulis dalam mengimplementasikan pengujian automation adalah untuk meneliti peran teknik BDD dalam implementasi automation test pada aplikasi GA System 2 United Tractors, serta membandingkan metode pengujian automation test dan manual test.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini adalah termasuk penelitian terapan dimana penelitian ini merupakan jenis penelitian yang dimaksudkan untuk menyelesaikan sebuah permasalahan dalam sebuah organisasi [8]. Gambar 1 menunjukkan ilustrasi tahapan penelitian. Dalam penelitian ini, diawali dengan analisa fitur beserta functional flow, memahami alur atau urutan langkah yang harus dilakukan dalam pengujian melalui dokumentasi dengan melakukan pengujian secara manual terhadap fitur



yang ada pada aplikasi. Tahap selanjutnya perancangan skenario pengujian, pada tahap tersebut skenario pengujian akan dijadikan landasan dalam pengujian masing masing fitur. Kemudian skenario pengujian akan di rancang lebih detail dengan pembuatan test case. Test case dibuat dengan dua jenis, yaitu negative test dan positive test. Kemudian tahap selanjutnya implementasi teknik bdd pada pembuatan feature file cucumber dan scripts groovy pada katalon studio. Setelah itu dilanjutkan dengan pembuatan test suite dan melakukan pengujian di katalon studio.

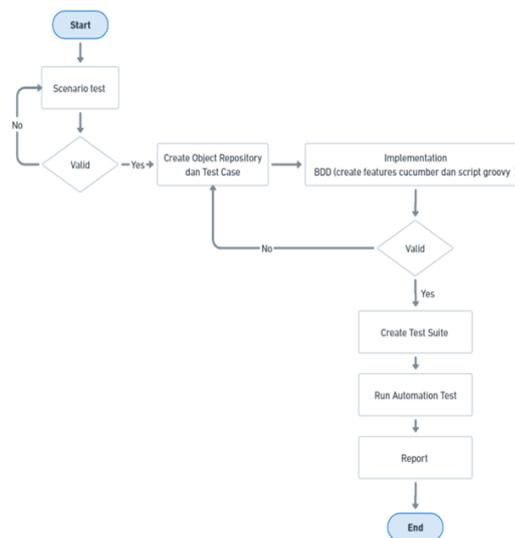


Gambar 1. Tahapan Penelitian

2.2 Behavior Driven Development

Behavior Driven Development atau BDD merupakan metode testing yang berfokus pada software behavior dengan cara mengilustrasikan behaviour sistem. Behavior yang dimaksud disini adalah penulisan skenario feature file sebelum implementasi pada test script. Behavior Driven Development (BDD) adalah metodologi agile yang diadopsi secara luas untuk pengembangan perangkat lunak yang menekankan perilaku aplikasi sebagai serangkaian kasus uji, menggunakan kata kunci, yang mencakup “given”, “when”, dan “then”. Hal ini melibatkan penulisan persyaratan dalam format terstruktur dan dapat diuji yang dapat dievaluasi untuk memastikan kepatuhan dengan perilaku yang diharapkan [10]. BDD didasarkan pada pendekatan yang digunakan dalam Test-Driven Development (TDD). Namun, BDD beroperasi pada tingkat yang lebih tinggi dari TDD dan bisa saja dianggap sebagai penyempurnaan TDD karena mengalihkan fokus dari pengujian ke identifikasi perilaku yang diharapkan dari suatu sistem.

Gambar 2 menunjukkan tahapan BDD pada penelitian ini. Langkah - langkah implementasi bdd dimulai dengan perancangan scenario test dan menyusun test case positif dan negatif untuk mencakup berbagai kemungkinan pengujian. Selanjutnya, komponen-komponen fitur aplikasi diubah menjadi object repository untuk memudahkan perencanaan test case. Metode Behavior Driven Development (BDD) diterapkan dengan membuat fitur cucumber yang menggambarkan perilaku yang diharapkan, serta menulis script groovy berdasarkan scenario test dan test case. Setelah itu, test suite dibuat untuk menjalankan automation test secara efisien dan teratur setelah implementasi BDD sesuai harapan dan tahap terakhir dengan pengecekan hasil pengujian melalui log viewer katalon studio dengan melihat runtime, dan juga melakukan pengecekan pada history test execution sebagai hasil pengujian pada automation test.



Gambar 2. Tahapan Implementasi BDD



3. HASIL DAN PEMBAHASAN

3.1 Analisis Pengujian Manual

Aplikasi GA System 2 United Tractors adalah perangkat lunak berbasis website yang membantu pengguna dalam permintaan kebutuhan general affair dan project management pada proyek konstruksi. Aplikasi tersebut memiliki beberapa fitur, yaitu login, home, manage data, project request, design & development, tender, contracts, monitoring, payments, dan geolocation, sign out. Pada kondisi existing Aplikasi GA System 2 United Tractors dilakukan secara manual testing. Pengujian hanya pada fitur tender dalam Aplikasi GA System 2 United Tractors dengan pertimbangan bahwa fitur tender memiliki kepentingan yang tinggi dalam Aplikasi GA System 2 United Tractors dan perlu dilakukan pengujian yang komprehensif untuk memastikan kualitas fitur tersebut. Namun, pengujian perangkat lunak secara manual membutuhkan ketelitian dan waktu yang lama. Pengujian ini hanya dapat dilakukan secara berkala dan tidak efisien untuk pengujian yang berulang. Selain itu, pengujian manual tidak dapat menjamin penurunan kecacatan perangkat lunak secara signifikan.

3.2 Pembuatan Skenario Automation Testing

Tahap selanjutnya setelah analisa fitur dan mengetahui functional flow, adalah membuat skenario pengujian. Skenario ini akan menjadi acuan dalam implementasi automation testing. Tabel 1 menunjukkan detail skenario automation testing pada fitur tender dalam aplikasi GA System 2 UT dengan ekspektasi hasil tes positif. Jika skenario ini menghasilkan hasil yang sesuai dengan ekspektasi ini, maka tes dianggap positif. Artinya, fungsionalitas login pada aplikasi berjalan dengan baik sesuai dengan yang diharapkan. Automation testing membantu memastikan bahwa skenario seperti ini dapat diulang secara otomatis dengan akurasi yang konsisten, dan dapat mendeteksi masalah potensial lebih awal dalam siklus pengembangan perangkat lunak

Tabel 1. Skenario Automation Testing dengan Ekspektasi Hasil Positif

Skenario	Tahapan	Expected Positive Result
Menambah <i>data vendor</i> dengan data baru	<ol style="list-style-type: none"> Masuk ke menu <i>tender</i> Klik <i>button add vendor</i> Isi data <i>vendor</i> Klik <i>button</i> simpan 	Sistem menerima penyimpanan data <i>vendor</i> kemudian menampilkan pada halaman <i>tender</i>
Batal tambah <i>vendor</i>	<ol style="list-style-type: none"> Masuk ke menu <i>tender</i> Klik <i>button add vendor</i> Isi data <i>vendor</i> Klik <i>button cancel</i> 	Sistem tidak menyimpan data <i>vendor</i> dan kembali ke halaman <i>tender</i>
Melihat details <i>project tender</i>	<ol style="list-style-type: none"> Masuk ke menu <i>tender</i> Pilih salah satu <i>project tender</i> Klik <i>button option</i> Pilih <i>details</i> <i>Download</i> salah satu <i>document</i>. 	Sistem akan menampilkan <i>details project</i> dan <i>file</i> akan terunduh

Selanjutnya adalah pembuatan skenario automation testing dengan ekspektasi hasil tes negative. Tabel 2 menunjukkan skenario dan tahapan pengujian otomatis fitur tender dalam aplikasi GA System 2 UT dengan ekspektasi hasil tes negatif. Automation testing dengan ekspektasi hasil tes negatif penting dalam pengembangan perangkat lunak karena membantu dalam mengidentifikasi masalah yang dapat mengganggu pengalaman pengguna atau keamanan aplikasi, serta memberikan kesempatan untuk memperbaiki masalah tersebut sebelum perangkat lunak diluncurkan.

Tabel 2. Skenario Automation Testing dengan Ekspektasi Hasil Negatif

Skenario	Step to test	Expected Negative Result
Menambah <i>data vendor</i> dengan data tidak di isi	<ol style="list-style-type: none"> Masuk ke menu <i>tender</i> Klik <i>button add vendor</i> Kosongkan Isi data <i>vendor</i> Klik <i>button</i> simpan 	Sistem akan menolak dan menampilkan pesan “Harap di Isi”



Menambah *data vendor* dengan data yang telah terdaftar

1. Masuk ke menu *tender*
2. Klik *button add vendor*
3. Isi data *vendor* dengan data terdaftar
4. Klik *button simpan*

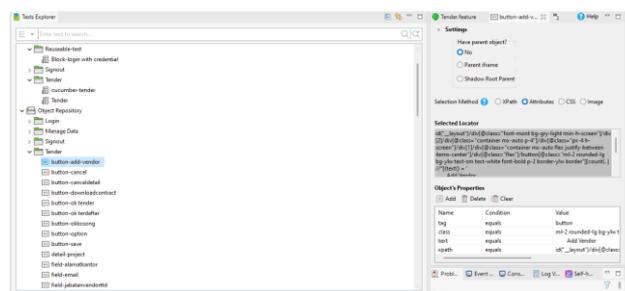
Sistem akan menolak dan menampilkan pesan “Data sudah terdaftar”

Pada Skenario Pengujian Fitur Tender (Positif Test). Positive test dilakukan dengan memakai parameter yang benar dan sesuai dengan tujuan fungsionalitas yang sedang diuji dan bertujuan menguji alur utama fungsi yang diuji. Sementara itu negative test dilakukan menggunakan parameter yang salah atau tidak sesuai dengan tujuan fungsionalitas yang diuji dan bertujuan untuk penanganan exception atau kondisi yang tidak valid dalam aplikasi. Pada skenario pengujian terdiri dari skenario, step to test dan expected result. Skenario menjelaskan tentang tindakan pengujian yang akan dilakukan. Step to test menjelaskan langkah - langkah pengujian sedangkan expected result menjelaskan tentang hasil yang diharapkan pada pengujian.

3.3 Pembuatan Test Case

Test case dalam software testing adalah sebuah dokumen atau langkah-langkah yang merinci skenario atau situasi pengujian yang harus dijalankan untuk memverifikasi atau memvalidasi suatu fitur, komponen, atau fungsi dalam perangkat lunak. Test case digunakan untuk menjalankan pengujian secara terstruktur, memastikan bahwa setiap fitur atau komponen perangkat lunak diperiksa sesuai dengan spesifikasi [11]. Hasil dari pengujian test case ini dapat membantu pengembang dan tim pengujian untuk memahami kualitas perangkat lunak, mengidentifikasi masalah, dan memastikan bahwa perangkat lunak berfungsi dengan baik sebelum diluncurkan ke pengguna akhir. Test case juga penting untuk mengulang pengujian di masa depan, memastikan bahwa perubahan atau perbaikan pada perangkat lunak tidak merusak fungsi yang sudah ada

Tahap pembuatan test case adalah dengan pengambilan object repository. Object repository merupakan objek yang di aplikasikan selama pengetesan atau sekumpulan test object. Pengambilan object repository dilakukan dengan menemukan locator object. Gambar 3 merupakan proses pengambilan object repository.



Gambar 3. Pengambilan Object Repository

Object repository adalah suatu struktur atau kumpulan elemen-elemen yang digunakan dalam pengujian perangkat lunak. Ini mencakup identifikasi dan deskripsi dari setiap elemen yang akan diuji, seperti tombol, input teks, tautan, atau elemen lainnya yang ada dalam aplikasi atau sistem yang akan diuji. Langkah Tahap selanjutnya adalah membuat test case pengujian. Pada tahap pembuatan test case, pengambilan object repository dilakukan akan digunakan dalam pengujian. Object repository ini berfungsi sebagai referensi sentral yang digunakan oleh tim pengujian untuk membangun dan menjalankan skenario pengujian. Dengan memiliki object repository yang jelas dan terstruktur, dapat dengan mudah mengakses dan menggunakan elemen-elemen yang diperlukan dalam pembuatan test case.

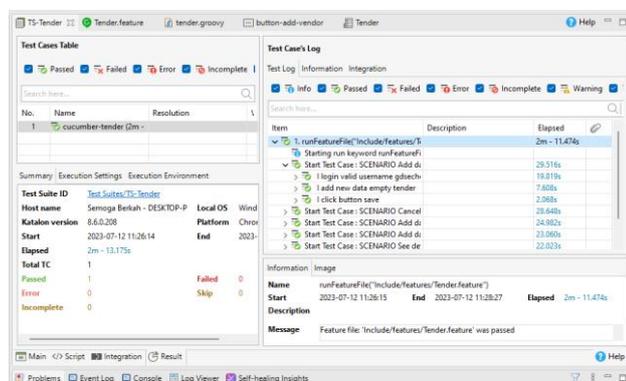


Item	Object	Input	Output
1 - Call Test Case	Block-login with credential	["username":"gdsrehead", "password":	
2 - Click	menu-tender		
3 - Click	button-add-vendor		
4 - Set Text	field-namakontraktor	namaKontraktor	
5 - Set Text	field-vendorpic	vendorPIC	
6 - Set Text	field-email	email	
7 - Set Text	field-alamat Kantor	"Jakarta"	
8 - Set Text	field-telepon	"087676554453"	
9 - Set Text	field-namavendorttd	"Gilang"	
10 - Set Text	field-jabatanvendorttd	"SPV"	
11 - Click	button-save		
12 - Click	button-ok tender		
13 - Click	button-okkosong		
14 - Click	button-cancel		
15 - Click	button-ok terddaftar		
16 - Click	button-option		
17 - Click	option-detail		
18 - Click	detail-project		
19 - Click	button-downloadcontract		
20 - Click	button-canceldetail		

Gambar 4. Pembuatan Test Case

3.4 Pembuatan Test Suite

Test suite merupakan kumpulan dari beberapa test case yang diatur sedemikian rupa untuk dijalankan secara bersamaan. Test suite berfungsi sebagai entitas yang memungkinkan pengujian dilakukan secara bersamaan. Dengan menggunakan test suite, dapat mengatur dan mengelompokkan test case berdasarkan fitur atau fungsi yang diuji secara otomatis dan berurutan.



Gambar 5. Pembuatan Test Suite

Gambar 5 menunjukkan pembuatan test suite aplikasi GA 2 System UT. Pada pembuatan Test suite berisi test case dan data file yang telah dibuat sebelumnya. Proses tersebut akan dijalankan serta melakukan identifikasi pencocokan antara variabel test case dengan data file.

3.5 Implementasi Teknik BDD

Behavior Driven Development (BDD) adalah pendekatan dalam pengembangan perangkat lunak yang berfokus pada perilaku atau keinginan pengguna [12]. Pendekatan ini menggabungkan elemen-elemen dari pengujian fungsional, analisis kebutuhan, dan kolaborasi antara pemangku kepentingan. Pengembangan dengan BDD dimulai dengan mendefinisikan perilaku yang diinginkan dari perangkat lunak, yang dinyatakan dalam bentuk cerita atau skenario yang dapat dibaca oleh semua pemangku kepentingan. Kemudian, tim pengembang dan pemangku kepentingan bekerja sama untuk mengimplementasikan perilaku tersebut melalui serangkaian tes yang terintegrasi dalam siklus pengembangan.

Dalam implementasi BDD, menulis skenario-skenario yang menggambarkan perilaku sistem dari perspektif pengguna. Skenario ditulis dalam format bahasa yang mudah dipahami oleh semua pihak yang terlibat dalam pengembangan sistem. Hal ini bertujuan untuk memastikan bahwa pemahaman bersama terhadap masalah yang harus dipecahkan terbangun di antara lintas peran terlibat. Terdapat 3 keyword utama dalam penulisan skenario BDD yaitu Given, When dan Then. Given menjelaskan konteks dari skenario, When menjelaskan tindakan yang dilakukan oleh user.

Gambar 6 menunjukkan pengaturan fitur file cucumber. Pada pengujian automation dan pendekatan Behavior Driven Development (BDD), yaitu feature file cucumber dan script groovy. Feature file cucumber adalah file teks yang digunakan untuk mendefinisikan perilaku atau skenario yang diharapkan dari sistem yang akan diuji. Feature file ini ditulis dalam bahasa yang mudah dibaca oleh manusia dan menggunakan sintaks Gherkin. Gherkin adalah bahasa yang digunakan



dalam BDD untuk menyatakan perilaku dengan format yang jelas dan terstruktur. Feature file berfungsi sebagai dokumen spesifikasi yang dapat dibaca oleh semua pemangku kepentingan, termasuk pengembang, tester, dan pemilik produk.

```
TS-Tender Tender.feature tender.groovy button-add-vendor Tender cucumber-tender
19 @tag
20 Feature: Tender
21
22 @tag1
23 Scenario Outline: Add data empty
24   Given I login valid username <username> and password <password>
25   When I add new data empty tender
26   Then I click button save
27
28 Examples:
29   | username | password |
30   | gdsehead | 123qwe* |
31
32 Scenario Outline: Cancel add data tender
33   Given I login valid username <username> and password <password>
34   When I click add data
35   Then I click button cancel
36
37 Examples:
38   | username | password |
39   | gdsehead | 123qwe* |
40
41 Scenario Outline: Add data new tender
42   Given I login valid username <username> and password <password>
43   When I add new data tender
44   Then I click save data
45
46 Examples:
47   | username | password |
48   | gdsehead | 123qwe* |
49
```

Gambar 6. Fitur File Cucumber

Script groovy adalah bahasa pemrograman yang digunakan dalam implementasi pengujian automation dengan alat pengujian seperti Cucumber. Gambar 7 merupakan tampilan script goovy. Groovy memberikan fleksibilitas dan kekuatan pemrograman untuk menulis skrip pengujian yang kompleks dan terstruktur. Dalam konteks BDD, Script groovy digunakan untuk mengimplementasikan perilaku yang dinyatakan dalam feature file cucumber. Skrip Groovy ini berfungsi sebagai penghubung antara spesifikasi yang dinyatakan dalam bahasa manusia dengan implementasi teknis dalam bahasa pemrograman.

```
TS-Tender Tender.feature tender.groovy button-add-vendor Tender cucumber-tender
400 /**
401  * The step definitions below match with Katalon sample @herkin steps
402  */
403
404 @Given("I login valid username (.*) and password (.*)")
405 def login(String username, String password) {
406   WEBUI.click(findTestObject("Newsekuar-test/block-login with credential"), [{"username": username, ("password": password}],
407   FailureHandling.STOP_ON_FAILURE)
408 }
409
410 @When("I add new data empty tender")
411 def AddTender() {
412   WEBUI.click(findTestObject("Tender/menu-tender"))
413   WEBUI.click(findTestObject("Tender/button-add-vendor"))
414 }
415
416 @Then("I click button save")
417 def Save() {
418   WEBUI.click(findTestObject("Tender/button-save"))
419   WEBUI.click(findTestObject("Tender/button-oksekuar"))
420 }
421
422 @When("I click add data")
423 def Clickdata() {
424   WEBUI.click(findTestObject("Tender/menu-tender"))
425   WEBUI.click(findTestObject("Tender/button-add-vendor"))
426 }
427
428 @Then("I click button cancel")
429 def Canceldata() {
430   WEBUI.click(findTestObject("Tender/button-cancel"))
431 }
432
```

Gambar 7. Scripts Groovy

3.6 Verifikasi Hasil Pengujian

Proses verifikasi hasil pengujian dapat dilakukan dengan menjalankan Test suite yang telah dibuat. Kemudian jumlah pengujian terdiri dari jumlah test case pengujian. Setelah pengujian dijalankan maka hasilnya dapat dilihat pada bagian log viewer seperti Gambar 8.

```
Run: 1/1 Passed: 1 Failures: 0 Errors: 0 Skips: 0
07-13-2023 09:12:33 PM Test Suites/TS-Tender
Elapsed time: 2s - 8.155s
Test Suite/TS-Tender (131.148)
  hostName = Sempaja Bekah - DESKTOP-F3JC1M0
  os = Windows 10.0.22000
  hostAddress = 192.168.0.102
  buildVersion = 3.6.2.0
  test Cases/Tender/cucumber-tender (127.477)
  1 - runFeatureDef('include/Features/Tender.feature') (126.591)
    Starting non keyword nonFeatureDef: 'include/Features/Tender.feature' and extract req
    ✓ SCENARIO Add data empty (26.624)
      1 - I login valid username gdsehead and password 123qwe* (20.598)
      2 - I add new data empty tender (6.026)
      3 - I click button save (0.224)
    ✓ SCENARIO Cancel add data tender (21.206)
      1 - I login valid username gdsehead and password 123qwe* (8.438)
      2 - I click add data (12.224)
      3 - I click button cancel (0.518)
    ✓ SCENARIO Add data new tender (28.548)
      1 - I login valid username gdsehead and password 123qwe* (16.544)
      2 - I add new data tender (5.404)
      3 - I click save data (2.224)
    ✓ SCENARIO Add data tender with registered data (22.318)
      1 - I login valid username gdsehead and password 123qwe* (8.544)
      2 - I add data tender with registered data (13.592)
      3 - I click save data registered (2.176)
    ✓ SCENARIO See detailed data tender (13.126)
```

Gambar 8. Log Viewer Test Suite

Dari log viewer dapat dilihat bahwa pengujian yang dilakukan berjalan sesuai dengan skenario yang telah dibuat. Ditandai



dengan indikator warna hijau. Berdasarkan hasil pengujian, dapat dilihat juga bahwa pengujian dengan skenario pengujian awal dan tidak menemukan bug. Apabila dibandingkan dengan pengujian manual. Maka hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Hasil Perbandingan antara Pengujian Otomatis dengan Manual

Pengujian	Keterangan	Waktu
Automation	Passed	4 menit
Manual	Passed	15 menit

Hasil perbandingan antara manual testing dan automation testing pada Aplikasi GA System 2 United Tractors akan memberikan pemahaman tentang kelebihan dan kekurangan masing-masing metode pengujian. Hal ini membantu menentukan pendekatan yang lebih efektif dan efisien dalam memastikan kualitas fitur tender dan aplikasi secara keseluruhan.

Pengujian manual pada fitur tender dalam Aplikasi GA System 2 United Tractors memakan waktu lama dan tidak efisien. Pengujian ini dilakukan secara berkala dan tidak menjamin penurunan kecacatan yang signifikan. Selain itu, manual testing tidak memungkinkan pengujian komprehensif pada fitur tender yang penting. Oleh karena itu, automation testing menjadi solusi yang efektif dan efisien. Dengan automation testing, pengujian fitur tender dapat dilakukan otomatis dan berulang dengan cepat. Proses pengujian dapat dilakukan secara sering dan terjadwal, meningkatkan efisiensi dan akurasi.

4. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa penggunaan automation testing dengan teknik behavior driven development (BDD) pada Aplikasi GA System 2 United Tractors telah terbukti efektif dan efisien dalam pengujian perangkat lunak. Hasil komparasi antara manual testing dan automation testing menunjukkan bahwa automation testing memberikan keuntungan dalam hal waktu, efisiensi, dan akurasi pengujian fitur tender. Implementasi automation testing menggunakan teknik BDD pada Aplikasi GA System 2 United Tractors memberikan kemudahan dalam memahami dan menjalankan pengujian dengan mudah. Selain itu, hasil pengujian automation menunjukkan rasio waktu yang lebih baik, dengan perbandingan 4:15 menit pada fitur tender dalam hal waktu pengujian dibandingkan dengan manual testing. Dalam skala project besar, pengujian automation dapat memberikan hasil yang memuaskan.

Penerapan automation testing dengan teknik BDD pada Aplikasi GA System 2 United Tractors memberikan manfaat signifikan dalam mempercepat proses pengujian, meningkatkan efisiensi pengembangan aplikasi, serta mengurangi risiko keberadaan bug pada saat rilis kepada pengguna. Hasil penelitian ini membuktikan bahwa pengujian otomatis dengan pendekatan BDD merupakan solusi yang efektif dalam meningkatkan kualitas dan keseluruhan pengujian perangkat lunak.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada LPPM STIKI Malang yang telah mendukung pelaksanaan penelitian ini. Selain itu penulis juga ingin menyampaikan terima kasih kepada tim developer aplikasi GA2 System United Tractor yang telah bersedia menjadi rekan peneliti selama proses penelitian

DAFTAR PUSTAKA

- [1] S. N. Dhuha, "Dictio Community," 2018. [Online]. Available: <https://www.dictio.idtapa-yang-dimaksud-dengan-pengujian-perangkatlunak-atau-software-testing/15136>.
- [2] H. N. Hadi, A. Aditya, F. E. Purwiantono and S. W. P. Listio, "PENGUJIAN PERFORMA PADA WEBSITE LOMBA NASIONAL KREATIVITAS MAHASISWA," *Jurnal Informatika*, vol. 22, no. 01, pp. 100-110, June 2022.
- [3] D. A. S. Sugiono, A. Aditya, H. N. Hadi and F. E. Purwiantono, "Analisis Usability Testing Guna Meningkatkan Efisiensi Kerja (Studi Kasus: Website Lo Kreatif)," in *Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK)*, Malang, 2022.
- [4] I. Kusyadi, S. Mulyati, A. P. Setiany, D. Noviyanto and S. Aisah, "Pengujian Aplikasi Kas Keuangan Menggunakan



- Katalon," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 2, pp. 91-96, 2022.
- [5] A. Zulianto, A. Purbasari, N. Suryani, A. I. Susanti, F. R. Rinawan and W. G. Purnama, "Pemanfaatan Katalon Studio untuk Otomatisasi Pengujian Black-Box pada Aplikasi iPosyandu," *JEPIN*, vol. 7, no. 3, pp. 370-378, 2021.
- [6] Y. Kosasih and A. B. Cahyono, "Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale," *AUTOMATA*, vol. 2, no. 1, 2021.
- [7] H. Rusli, "Analisa Perbandingan Black-Box Automated Testing dan Manual Testing pada Aplikasi ACCMART," *Skripsi*, 2020.
- [8] L. Hakim, S. P. Kristanto, M. N. Shodiq and E. Amaliyah, "APLIKASI PENERIMAAN DAN PENGELUARAN KAS BERBASIS WEB DAN WHATSAPP GATEWAY," *Jurnal Teknokompak*, vol. 15, no. 1, pp. 13-25, 2021.
- [9] A. Aditya, S. Aminah and Y. A. Kanthi, *Metodologi Penelitian Dalam Disipilin Ilmu Sistem Informasi*, Yogyakarta: Penerbit Andi, 2022.
- [10] M. S. Farooq, U. Omer, A. Ramzan, M. A. Rasheed and Z. Atal, "Behavior Driven Development: A Systematic Literature Review," *IEEE Access*, pp. 88008 - 88024, 2023.
- [11] A. Fahrezi, F. N. Salam, G. M. Ibrahim, R. R. Syaiful and A. Saifudin, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia," *LOGIC : Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, no. 1, pp. 1-5, 2022.
- [12] S. N. Hijriyani, S. Widowati and D. S. Kusumo, "PENERAPAN BEHAVIOR DRIVEN DEVELOPMENT UNTUK UJI VALIDASI," *TELKATIKA*, vol. 2, no. 1, pp. 96-104, 2022.
- [13] T. A. Lamaluta, C. E. J. C. Montolalu, A. L. Kalua, M. I. Takaendengan, and V. Y. G. Montolalu, "Web-Based School Information System in Permata Hati Speacial School for Autism Manado," *J. Ilm. Inform. dan Ilmu Komput.*, vol. 2, no. 2, pp. 65-74, 2023.
- [14] I. A. Alfarisi, A. T. Priandika, and A. S. Puspaningrum, "Penerapan Framework Laravel Pada Sistem Pelayanan Kesehatan (Studi Kasus: Klinik Berkah Medical Center)," *J. Ilm. Comput. Sci.*, vol. 2, no. 1, pp. 1-9, 2023, doi: 10.58602/jics.v2i1.11.
- [15] A. Putra, M. R. D. Susanto, and Y. Fernando, "Penerapan MDLC Pada Pembelajaran Aksara Lampung Menggunakan Teknologi Augmented Reality," *Chain J. Comput. Technol. Comput. Eng. Informatics*, vol. 1, no. 2, pp. 32-43, 2023.