

Teknik Equivalence Partitioning Otomatis pada Pengujian Black Box: Studi Kasus pada API Loyalitas MyPertamina

Automatic Equivalence Partitioning Technique on Back Box Testing: A Case Study on MyPertamina Loyalty API

Siska Arifiani¹, Dias Tri Kurniasari², Siti Rochimah³
^{1,2,3}Teknik Informatika, Institut Teknologi Sepuluh Nopember
E-mail: siska@its.ac.id

Abstrak

Perkembangan teknologi pada aplikasi *smartphone* yang cukup pesat saat ini dapat membantu mempermudah aktivitas manusia. MyPertamina adalah aplikasi *mobile* yang ditawarkan oleh Pertamina untuk membantu menawarkan berbagai produk Pertamina secara *online*. Terdapat beberapa modul pada MyPertamina antara lain *payment*, *voucher*, *user*, dan *loyalty*. Modul *loyalty* menyediakan layanan terkait informasi keuntungan yang diperoleh para pengguna aplikasi MyPertamina. Pada penelitian ini akan dikembangkan mekanisme pengujian otomatis modul *loyalty* MyPertamina dengan menggunakan teknik pengujian *equivalence partitioning*.

MyPertamina dikembangkan dengan menggunakan RESTfull API. RESTful API merupakan sebuah mekanisme dimana aplikasi pada perangkat *client* (pengguna) mengakses *end point* yang disediakan untuk dapat menjalankan fitur tertentu. Untuk melakukan pengujian pada aplikasi yang memanfaatkan RESTful API, dibutuhkan skenario pengujian yang berbeda. Pengujian Black Box dengan menggunakan teknik *equivalence partitioning* pada RESTfull API MyPertamina diharapkan dapat membantu melakukan pengujian yang lebih efektif dan efisien. Pada penelitian ini, dilakukan perbandingan pengujian manual dan pengujian otomatis pada RESTful API MyPertamina. Pengujian manual dilakukan dengan menggunakan Postman dengan mencatat hasil pengujian dari masing-masing data uji yang digunakan. Sedangkan pengujian otomatis dijalankan menggunakan pytest. Hasil penelitian ini menunjukkan bahwa pengujian otomatis memiliki hasil akurasi yang baik, jika dibandingkan dengan pengujian manual. Selain itu, efisiensi waktu pengujian pada pengujian otomatis adalah lebih baik jika dibandingkan dengan pengujian manual.

Kata kunci: Equivalence Partitioning, MyPertamina, Pengujian Perangkat Lunak, RESTful API

Abstract

The rapid development of smartphone application technology has significantly facilitated human activities. MyPertamina, a mobile application offered by Pertamina, provides a platform for online purchasing of various Pertamina products. MyPertamina consists of several modules, including payment, voucher, user, and loyalty. The loyalty module offers services related to user benefits. This research aims to develop an automated testing mechanism for the MyPertamina loyalty module using the equivalence partitioning testing technique.

MyPertamina is developed using RESTful APIs, a mechanism where client-side applications access provided endpoints to execute specific features. Testing applications that utilize RESTful APIs requires distinct testing scenarios. Black box testing with equivalence partitioning on MyPertamina's RESTful API is expected to enhance testing efficiency and effectiveness. This study compares manual and automated testing of MyPertamina's RESTful API. Manual testing is conducted using Postman, recording the results for each test data, while automated testing is executed using pytest. The results demonstrate that automated testing offers higher accuracy compared to manual testing. Additionally, automated testing exhibits better time efficiency.

Keywords: Equivalence Partitioning, MyPertamina, Software Testing, RESTful API

1. PENDAHULUAN

Smartphone adalah telepon genggam yang memiliki kemampuan tinggi dan dilengkapi dengan berbagai fitur yang mutakhir [1]. Perkembangan *smartphone* pada saat ini semakin pesat yang disebabkan oleh adanya berbagai aplikasi *mobile* yang ditawarkan dalam mempermudah aktivitas manusia. Pada kuartal ke-1 tahun 2022, Statista mencatat jumlah aplikasi yang tersedia pada Google Play Store mencapai 3,29 juta sedangkan Apple's App Store hadir dengan lebih dari 2,11 juta aplikasi yang tersedia [2]. Hal ini menunjukkan bahwa pengguna *smartphone* tidak dapat terlepas dari penggunaan aplikasi yang mereka unduh secara gratis maupun berbayar untuk memudahkan berbagai macam hal di kehidupan sehari-hari. Saat ini dalam pengembangan perangkat lunak baik berbasis *mobile* ataupun web banyak menggunakan API dikarenakan penggunaannya yang lebih mudah dan dapat diintegrasikan ke berbagai aplikasi. API (Application Programming Interface) adalah sebuah interface yang menghubungkan suatu aplikasi dengan aplikasi lainnya [3]. Dengan adanya API berbagai perangkat lunak yang berbeda, baik dalam satu platform yang sama atau pun lintas platform dapat terhubung. Penerapan API mempermudah pengembang dalam melakukan perbaikan atau perubahan sistem dikarenakan proses logika terhadap basis data dengan logika terhadap tampilan antarmuka menjadi suatu program yang terpisah. Selain itu, API dapat digunakan untuk komunikasi dengan berbagai bahasa pemrograman yang berbeda dan tidak perlu menyediakan data sendiri karena cukup mengambil data yang dibutuhkan dari platform lain [3].

Dalam proses pengembangan perangkat lunak dengan menggunakan RESTful API, perlu dilakukan pengujian terhadap RESTful API yang telah dibuat untuk meminimalisir kesalahan yang ada pada RESTful API tersebut dan dapat segera diperbaiki. Selain itu, pengujian setelah pembuatan berguna untuk memastikan kualitas dan performa dari RESTful API yang dibuat. Terdapat berbagai metode pengujian perangkat lunak salah satunya adalah Black Box testing. Pengujian pada tampilan aplikasi, fungsi- fungsi, dan kesesuaian alur fungsi dengan proses bisnis merupakan mekanisme pada pengujian Black Box. Black Box testing berfokus pada keluaran yang dihasilkan oleh sistem dan tidak melakukan pengujian pada *source code*. Pada Black Box testing terdapat beberapa mekanisme pengujian salah satunya adalah *equivalence partitioning testing*. *Equivalence partitioning testing* merupakan mekanisme pengujian Black Box dengan menguji kesesuaian antara masukan data dan keluaran data [4]. Penerapan Black Box testing diharapkan dapat membantu pengembang (*tester*) dalam melakukan pengujian dan memastikan perangkat lunak yang dibuat sesuai dengan harapan atau persyaratan yang telah disetujui. MyPertamina adalah sebuah aplikasi *mobile* yang digunakan untuk membeli berbagai produk yang ditawarkan oleh Pertamina secara *online* yang dapat dilakukan kapan pun dengan berbagai jangkauan area pembelian. Aplikasi ini dilengkapi dengan berbagai fitur diantaranya Pertamina Delivery Service, pembelian produk BBM ataupun *voucher*, informasi Pertamina, dan fitur-fitur pendukung lainnya. Dalam pengembangannya terbagi menjadi beberapa modul yaitu payment, voucher, user, dan loyalty. Salah satu modul yang cukup penting adalah modul loyalty yang berisi fitur- fitur tentang keuntungan pengguna. Pengembangan aplikasi MyPertamina secara keseluruhan menggunakan RESTful API.

Pengujian perangkat lunak merupakan hal yang sering digunakan untuk memverifikasi dan memvalidasi kualitas dari sebuah perangkat lunak [5]. Pengujian pada perangkat lunak dapat dilakukan dari awal pengembangannya. Pengujian yang dilakukan dari awal dapat menurunkan resiko kesalahan akhir aplikasi yang dibuat. Pengujian dapat dilakukan secara berkala dari setiap tahapan yang telah selesai dikerjakan. Salah satu yang dapat dilakukan pengujian di awal adalah RESTful API yang telah selesai dibuat. RESTful API merupakan bagian yang sangat penting dalam pengembangan aplikasi sebelum keseluruhan komponen diintegrasikan dalam fase produksi. Oleh karena itu, pada penelitian ini peneliti akan membahas mengenai "Implementasi Black Box Testing pada pengembangan RESTful API aplikasi *mobile* MyPertamina modul loyalty dengan teknik *equivalence partitioning*". Pada penelitian ini, peneliti menerapkan metode Black Box testing dengan teknik *equivalence partitioning* pada RESTful API yang telah dibuat untuk mengetahui kesesuaian antara API yang dibuat dengan yang diharapkan oleh pengembang.

Pengujian ini dilakukan menggunakan dua metode yaitu secara manual menggunakan postman dan otomatis menggunakan pytest. Penelitian ini diharapkan dapat diperoleh hasil pengujian yang baik sehingga meminimalisir *bug* yang ditemukan dalam MyPertamina pada fase production. Pengujian ini dilakukan untuk memastikan bahwa kualitas dan performa RESTful API yang sudah dibuat dapat berjalan dengan baik.

2. PENGUJIAN RESTFUL API DENGAN TEKNIK EQUIVALENCE PARTITIONING

Menurut standar ANSI/IEEE 1059, pengujian perangkat lunak adalah suatu proses menganalisis perangkat lunak yang bertujuan untuk mendeteksi perbedaan antara hasil yang ada dengan yang diinginkan (*error*) dan mengevaluasi fitur-fitur dari entitas perangkat lunak [6].

Application Programming Interface (API) adalah sekumpulan perintah, fungsi, dan protokol yang digunakan oleh seorang programmer ketika membangun atau mengembangkan suatu perangkat lunak [6]. Penggunaan API membuat pengembangan perangkat lunak menjadi lebih efisien dan program yang dibuat lebih mudah serta sederhana. API membantu dua program atau aplikasi untuk saling berkomunikasi dan menyediakan fungsi yang diperlukan. Proses jalannya API yaitu adanya *request* dari pengguna dan mengirimkan *request* tersebut ke *server* dan *server* akan mengirimkan hasil *request* kepada pengguna.

Representational State Transfer (REST) adalah salah satu desain arsitektur yang mendefinisikan serangkaian fungsi dalam membuat layanan web. REST API merupakan cara mengakses layanan web secara sederhana dan fleksibel tanpa perlu pemrosesan apa pun. Penggunaan REST lebih disukai daripada SOAP dikarenakan REST menggunakan *bandwidth* yang lebih sedikit, sederhana, dan fleksibel sehingga lebih cocok untuk pengguna internet. REST API digunakan untuk mengambil atau memberikan informasi dari suatu layanan web dengan menggunakan HTTP *request*. Cara kerja dari RESTful API yaitu REST klien akan melakukan akses pada data pada REST server masing-masing *resource*. Kemudian, REST server akan mengirimkan response yang dapat berupa HTML, XML, gambar, atau JSON yang saat ini paling populer digunakan [7].

Black Box testing merupakan pelengkap dari metode White Box. Hal ini dikarenakan kesalahan yang tidak didapatkan dengan metode White Box dapat ditemukan pada penerapan metode Black Box testing. Black Box testing adalah pengujian yang berfokus pada detail perangkat lunak seperti fungsi-fungsi dalam perangkat lunak, proses bisnis, dan tampilan perangkat lunak. Dalam melakukan pengujian dengan Black Box hanya mengecek nilai dari masing-masing masukan dan tidak ada langkah untuk mencari kode program yang digunakan untuk *output*. Keuntungan dalam menggunakan metode Black Box adalah seorang penguji tidak perlu memiliki pengetahuan dalam tentang pemrograman [8]. Sudut pandang pengguna menjadi acuan dalam pengujian Black Box. Pada metode Black Box terdapat 3 teknik yang dapat digunakan yaitu pengujian *graph-based*, *equivalence partitioning*, dan *boundary value analysis*.

Equivalence Partitioning merupakan salah satu teknik pengujian dalam metode Black Box testing. Equivalence Partitioning adalah mekanisme pengujian yang membagi atau mengelompokkan *input* dari suatu program ke dalam kelas data, menentukan kasus pengujian dalam menemukan kelas-kelas kesalahan, sehingga akan mengurangi jumlah keseluruhan kasus pengujian. Kelas ekuivalensi merepresentasikan serangkaian kondisi valid dan invalid untuk kondisi masukan. Jenis masukan data pada form dapat berupa numerik, suatu rentang harga, serangkaian harga yang terkait, atau teks.

Postman merupakan sebuah aplikasi yang digunakan untuk melakukan pengujian terhadap REST API. Postman banyak digunakan untuk pengujian aplikasi agar lebih baik oleh seorang *quality assurance* dan pengembang. Postman mengirimkan permintaan API ke server web dan menampilkan *response* dari API yang dibuat [9].

Pada penelitian ini, digunakan Pytest untuk mengembangkan sebuah teknik pengujian otomatis. Pytest itu sendiri adalah framework *automation test* dalam bahasa pemrograman python. Pytest dapat digunakan pada berbagai tipe dan level dalam *software testing*. Penggunaan pytest dapat mempersingkat waktu pengujian. Hal ini dikarenakan pengujian dapat dilakukan

pada satu folder yang berisi banyak file *script* yang perlu diuji sehingga kita tidak perlu menjalankan file *script* secara terpisah. Agar suatu folder, file, class, maupun method dapat diuji secara otomatis oleh Pytest dalam penamaannya harus mengandung kata test misalnya test_pertambahan [10].

Beberapa penelitian sebelumnya yang dijadikan rujukan pada penelitian ini dapat dideskripsikan pada Tabel 1. Berdasarkan hasil uraian tersebut dapat disimpulkan bahwa mayoritas penelitian berfokus pada antarmuka (UI) sistem dan fitur atau fungsionalitas sistem berbasis *web* dan *mobile*. Pengujian masih dilakukan secara manual. Terdapat sedikit penelitian yang berfokus pada RESTful API dengan teknik *equivalence partition*. Penelitian ini memiliki kontribusi dalam melakukan pengujian RESTful API pada aplikasi MyPertamina dengan menggunakan teknik *equivalence partitioning* secara otomatis menggunakan Python.

Tabel 1. Perbandingan Penelitian Sebelumnya

No	Judul	Kekurangan	Kelebihan
1	Implementasi Automated Functional UI Testing Untuk Mencapai Konsep CI/CD Terintegrasi Jenkins Menggunakan Framework Serenity dengan Page Object Model Design Pattern Pada PPDB Jatim [11]	Sebatas UI Testing dan perlu menggabungkan beberapa teknik pengujian untuk mendapatkan hasil pengujian yang lebih dalam	Pengujian otomatis menggunakan konsep CI/CD berjalan lebih cepat dibandingkan manual dan menghasilkan laporan yang terstruktur
2	Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web [12]	Pengujian masih dilakukan manual dengan membagi ruang pilihan skenario dan data pengujian	Pengujian yang dilakukan dengan Teknik <i>equivalence partitioning</i> paling sesuai digunakan karena menguji sistem dengan membagi ruang menjadi pilihan. Dan dari hasil pengujian yang telah dilakukan, sistem dapat digunakan dengan baik dan sempurna.
3	Implementasi metode pengujian equivalence partitioning pada pengembangan RESTful API Sistem Informasi Klinik Pratama UPN “Veteran” Yogyakarta [13]	Pengujian <i>equivalence partitioning</i> masih dilakukan secara manual menggunakan Postman. Data uji ditambahkan melalui Postman sesuai format yang disediakan oleh <i>endpoint</i> RESTful API yang disediakan.	RESTful API Klinik Pratama UPN Veteran Yogyakarta berjalan sesuai tujuan yang diharapkan yang dibuktikan dengan tidak ada kesalahan yang terjadi pada sistem ketika dilakukan pengujian dengan menggunakan teknik <i>equivalence partitioning</i> .
4	Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods [14]	Masing-masing teknik memiliki kekurangan dan kelebihan. Tingkat jangkauan dari data uji beragam, bergantung teknik yang digunakan dan	Pengujian dengan teknik boundary value analysis digunakan untuk menganalisis nilai batas sesuai kebutuhan seperti nomor KTP. Sedangkan

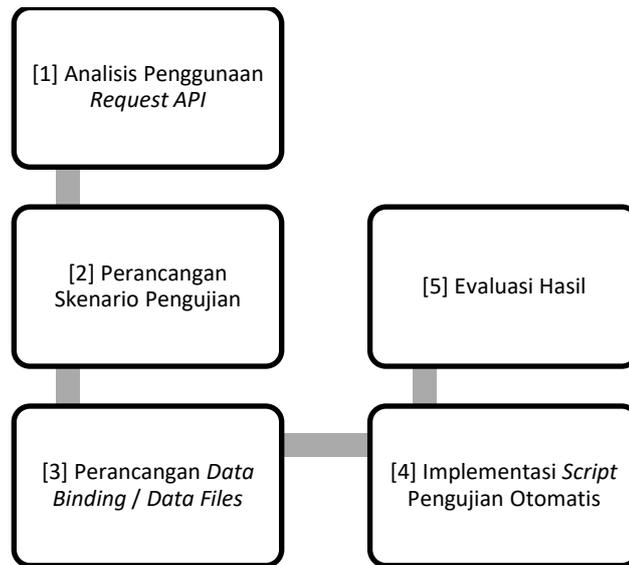
		rancangan skenario pengujian. Teknik <i>equivalence partition</i> dan Boundary Value Analysis (BVA) dilakukan secara manual dengan membentuk skenario uji dan data uji yang sesuai	pengujian dengan teknik <i>equivalence partitioning</i> digunakan untuk menguji fungsionalitas masukan data dan validasi
5	Black Box Testing with Equivalence Partitions Techniques in Transcrop Applications [15]	Masih manual dalam melakukan pengujian. Selain itu Teknik <i>equivalence partition</i> memiliki hasil yang kurang optimal jika dibandingkan dengan metode seperti BVA. Sehingga perlu menggabungkan beberapa teknik agar lebih optimal	Penelitian ini menggunakan dokumen <i>software requirement specification</i> (SRS) untuk dapat membentuk skenario uji dan data uji. Pengujian dilakukan sebelum perangkat lunak dikembangkan. Hasil pengujian selanjutnya akan dijadikan referensi untuk memperbaiki perangkat lunak
6	E-Learning Evaluation of Del Superior High School Based on Black Box Testing with Equivalence Partitioning and Graph-Based Testing [16]	Pengujian manual yang dilakukan hanya berfokus pada kebutuhan fungsional. Untuk pengembangan selanjutnya, modifikasi pada teknik pengujian adalah dibutuhkan untuk meningkatkan kualitas dari sistem yang dapat mencakup kebutuhan fungsional dan non-fungsional.	Penelitian ini menggunakan teknik <i>equivalence partitioning</i> dan <i>graph-based testing</i> untuk membuat sebuah skenario pengujian. Hasil pengujian yang dilakukan dapat membantu dokumentasi dan perbaikan sistem E-Learning selanjutnya.

3. METODOLOGI

Penelitian ini dilakukan dengan melewati beberapa tahap yang dimulai dengan membaca dan menganalisis *manual guide* dari aplikasi MyPertamina beserta RESTful API. Pada *manual guide* terdapat persyaratan yang harus dipenuhi dari setiap fitur API yang ada, aksi yang harus dilakukan, dan hasil keluaran yang diharapkan. Setelah dilakukan analisis terhadap *manual guide*, langkah yang akan dilakukan selanjutnya adalah membuat skenario pengujian dengan melakukan pengelompokan masukan data berdasarkan fungsinya. Kemudian, membuat data file atau data binding yang akan digunakan selama pengujian. Data files dapat berupa input, harapan pengujian, dan hal lain yang diperlukan selama pengujian berlangsung.

Langkah selanjutnya adalah pembuatan *script* pengujian. *Script* pengujian dibuat menggunakan bahasa pemrograman python. Tahapan akhir dari penelitian ini adalah melakukan pengujian terhadap RESTful API baik secara manual maupun otomatis. Pada pengujian secara manual dilakukan menggunakan Postman sesuai dengan skenario pengujian dan data binding yang telah dibuat. Sedangkan untuk pengujian secara otomatis dilakukan dengan menggunakan pytest untuk melakukan pengujian pada *script* yang telah dibuat. Uraian tahapan pada penelitian

ini dapat ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Pengujian Aplikasi MyPertamina

4. HASIL DAN PEMBAHASAN

4.1. Analisis Manual Guide

Pada penelitian ini sistem yang digunakan sebagai bahan uji adalah API aplikasi MyPertamina modul *loyalty*. Aplikasi MyPertamina dibangun menggunakan API oleh beberapa *developer*. Aplikasi MyPertamina berfokus pada *loyalti* yang diberikan kepada pengguna BBM. Aplikasi ini digunakan untuk melakukan pembelian BBM, pembelian produk Pertamina, dan penukaran poin. Pada modul *loyalty* sendiri fokus utamanya adalah *reward* yang diberikan setelah pembelian produk.

Fitur pada aplikasi MyPertamina meliputi kegiatan pembelian produk Pertamina. Pada penelitian ini, fitur yang dibahas hanya pada modul *loyalti*. Hal ini dikarenakan, tujuan utama adanya aplikasi ini adalah untuk memberikan penghargaan kepada masyarakat yang telah membeli produk Pertamina menggunakan aplikasi. Penghargaan yang diberikan berupa poin yang dapat digunakan untuk pembelian produk lain yang tersedia di aplikasi MyPertamina. Sehingga fokus utama untuk fitur yang diuji adalah pemberian dan klaim poin serta pengurangan poin. Fitur-fitur tersebut diantaranya *claim point*, *coin adjustment*, dan *bulk inject point*.

4.2. Pembuatan Skenario Pengujian

Setelah memahami *manual guide*, maka hal yang dilakukan selanjutnya adalah pembuatan skenario pengujian. Skenario pengujian adalah rancangan keadaan yang akan diujikan pada API untuk memastikan fungsionalitas program apakah sudah sesuai dengan kebutuhan yang diinginkan atau masih terdapat beberapa kesalahan atau kesalahan yang terjadi sehingga masih dibutuhkannya perbaikan. Metode yang digunakan dalam membuat skenario pengujian fungsionalitas adalah metode Black Box dengan teknik *equivalence partitioning*. Metode ini berfokus pada nilai keluaran yang diberikan oleh nilai masukan maupun terhadap proses yang terjadi. Skenario pengujian lebih jelasnya dapat dilihat pada Tabel 2 hingga Tabel 9.

Tabel 2. Skenario pengujian API Check Estimate

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. <i>User check estimate</i> dengan memasukkan jumlah pembelian lebih dari sama dengan 0.	POST	Data A01 – A05	<i>Check estimate</i> pembelian produk berhasil.
02	Invalid. <i>User check estimate</i> dengan memasukkan jumlah pembelian kurang dari 0.	POST	Data A06 – A10	<i>Check estimate</i> pembelian produk gagal dan muncul pesan kesalahan jumlah pembelian harus bilangan positif.
03	Invalid. <i>User check estimate</i> untuk produk yang tidak ada.	POST	Data A11 – A15	<i>Check estimate</i> pembelian produk gagal dan muncul pesan kesalahan rule produk tidak ditemukan.
04	Valid. <i>User check estimate</i> tanpa mengisi jumlah pembelian.	POST	Data A16- A20	<i>Check estimate</i> pembelian produk berhasil.

Tabel 3. Skenario pengujian API Run Activity

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. <i>User run activity rule</i> dengan memasukkan jumlah pembelian lebih dari sama dengan 0.	POST	Data B01 – B05	<i>Run activity rule</i> pembelian produk berhasil.
02	Invalid. <i>User run activity rule</i> dengan memasukkan jumlah pembelian kurang dari 0.	POST	Data B06 – B10	<i>Run activity rule</i> pembelian produk gagal dan muncul pesan kesalahan jumlah pembelian harus bilangan positif.
03	Invalid. <i>User run activity rule</i> untuk produk yang tidak ada.	POST	Data B11 – B15	<i>Run activity rule</i> pembelian produk gagal dan muncul pesan kesalahan an produk tidak ditemukan.
04	Valid. <i>User run activity rule</i> tanpa mengisi jumlah pembelian.	POST	Data B16- B20	<i>Run activity rule</i> pembelian produk berhasil.
05	Invalid. <i>User run activity rule</i> dengan jumlah pembelian melebihi batas maksimum harian	POST	Data B21- B25	<i>Run activity rule</i> pembelian produk gagal dan muncul pesan kesalahan, telah mencapai limit harian.

Tabel 4. Skenario pengujian API Coin Adjustment

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. User menambahkan poin dengan memasukkan Nomor HP dan jumlah koin.	POST	Data C01 – C05	Penambahan poin berhasil dilakukan.
02	Valid. User mengurangi poin dengan memasukkan Nomor HP dan jumlah koin.	POST	Data C06 – C10	Pengurangan poin berhasil dilakukan.
03	Valid. User mengurangi atau menambah poin dengan memasukkan tipe masukan yang tidak sesuai.	POST	Data C11 – C15	Pengurangan atau penambahan poin gagal dilakukan, dan muncul kesalahan tipe masukan salah.
04	Invalid. User mengurangi poin dengan memasukkan Nomor HP yang tidak sesuai.	POST	Data C16 - C20	Pengurangan poin gagal dilakukan dan muncul pesan kesalahan, Nomor HP tidak ditemukan.
05	Invalid. User mengurangi poin dengan memasukkan jumlah poin kurang dari 0.	POST	Data C21 – C25	Pengurangan poin gagal dilakukan dan muncul pesan kesalahan, jumlah poin harus bilangan positif.
06	Invalid. User mengurangi poin melebihi poin yang dimiliki.	POST	Data C26 – C30	Pengurangan poin gagal dilakukan dan muncul pesan kesalahan, jumlah koin harus kurang atau sama dengan yang dimiliki.
07	Invalid. User mengurangi poin tanpa mengisi kolom isian atau isian tidak sesuai.	POST	Data C31- C40	Pengurangan poin gagal dilakukan dan muncul pesan kesalahan, user harus mengisi kolom isian Nomor HP dan jumlah koin.

Tabel 5. Skenario pengujian API Create Ticket Bulk Inject Poin

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. User create ticket dengan mengupload file CSV, memasukkan nama ticket dan tags.	POST	Data D01 - D05	Ticket berhasil dibuat dan ditambahkan dalam database.
02	Invalid. User create ticket dengan mengupload file selain CSV.	POST	Data D06- D10	Pembuatan ticket gagal dan muncul pesan kesalahan file harus berupa CSV.
03	Invalid. User create ticket tanpa memasukkan nama ticket.	POST	Data D11- D16	Pembuatan ticket gagal dan muncul pesan kesalahan, nama ticket harus diisi.
04	Invalid, User create ticket tanpa memasukkan tags.	POST	Data D16- D20	Pembuatan ticket gagal dan muncul pesan kesalahan, tags harus diisi.

Tabel 6. Skenario pengujian API Customer Check Point Estimation

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. <i>User check point estimation</i> dengan mengisi nama produk, jumlah, dan harga produk.	POST	Data E01-E05	Mendapatkan hasil estimasi poin.
02	Invalid. <i>User check point estimation</i> dengan mengisi nama produk yang tidak sesuai	POST	Data E06 – E10	Gagal mendapatkan hasil estimasi poin.
03	Invalid. <i>User check point estimation</i> tanpa mengisi nama produk	POST	Data E11-E15	Gagal mendapatkan hasil estimasi poin dan muncul pesan kesalahan, nama produk harus diisi.
04	Invalid. <i>User check point estimation</i> tanpa mengisi jumlah pembelian produk atau mengisi jumlah yang tidak sesuai	POST	Data E16 – E20	Gagal mendapatkan hasil estimasi poin dan muncul pesan kesalahan, jumlah pembelian harus diisi dan berupa angka.
05	Invalid. <i>User check point estimation</i> tanpa mengisi harga produk atau mengisi harga yang tidak sesuai	POST	Data E21 – E25	Gagal mendapatkan hasil estimasi poin dan muncul pesan kesalahan, harga produk harus diisi dan berupa angka.
06	Invalid. <i>User check point estimation</i> dengan memasukkan jumlah pembelian produk kurang dari 0.	POST	Data E26-E30	Hasil estimasi poin pembelian produk gagal didapatkan.

Tabel 7. Skenario pengujian API Get One Ticket Bulk Inject Poin

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. <i>User get one ticket</i> dengan memasukkan parameter id yang sesuai.	GET	Data F01 – F05	Mendapatkan detail tiket.
02	Invalid. <i>User get one ticket</i> dengan memasukkan parameter id yang tidak sesuai.	GET	Data F06 – F10	Gagal mendapatkan detail tiket dan muncul pesan kesalahan, tiket tidak ditemukan.

Tabel 8. Skenario pengujian API Get Url Ticket Bulk Inject Poin

Test Case ID	Detail Test Case	Method	Masukan	Hasil yang Diharapkan
01	Valid. <i>User get URL CSV dengan memasukkan parameter id yang sesuai.</i>	GET	Data G01 – G05	Mendapatkan URL CSV.
02	Invalid. <i>User get URL CSV dengan memasukkan parameter id yang tidak sesuai.</i>	GET	Data G06 – G10	Gagal mendapatkan URL CSV dan muncul pesan kesalahan, CSV tidak ditemukan.

Tabel 9. Deskripsi Tipe Data API

No	API	Masukan	Deskripsi
1.	<i>Check Estimate dan Run Activity Rule</i>	<i>Amount</i>	Jumlah pembelian produk memiliki tipe data angka. Pada pengujian <i>amount</i> dipilih beberapa angka yang termasuk dalam bilangan bulat positif dan beberapa angka bilangan negative.
		<i>Product</i>	Nama produk yang dibeli memiliki tipe data <i>string</i> . Pengujian produk dipilih beberapa kombinasi penulisan kata seperti lowercase, uppercase dan yang lainnya. Selain itu, dilakukan pengujian pada data produk yang tidak ada pada database.
2.	<i>Coin Adjustment</i>	<i>Type</i>	Tipe data pada <i>coin adjustment</i> adalah <i>string</i> . Tipe masukan terdiri dari 2 pilihan yaitu <i>add</i> dan <i>min</i> . akan tetapi, dalam pembuatan data uji juga dilakukan pengujian pada tipe data masukan yang berbeda atau tidak sesuai.
		<i>No Handphone</i>	Dalam request API, masukan nomor handphone memiliki tipe data <i>string</i> . Pengujian nomor handphone dipilih dari beberapa nomor telepon dengan jumlah yang sesuai 10-14 digit dan digit yang kurang atau lebih.
		<i>Amount</i>	Jumlah koin yang diberikan atau dikurangi bertipe data angka. Pada pengujian <i>amount</i> dipilih beberapa angka yang termasuk dalam bilangan

			positif dan beberapa angka bilangan negative. Selain itu, juga dilakukan pengujian kombinasi angka dan huruf.
3.	<i>Create Ticket Bulk Inject Point</i>	<i>ticketName</i>	Penamaan tiket yang akan dibuat memiliki tipe data <i>string</i> . Pengujian nama tiket dilakukan dengan memilih beberapa kombinasi penulisan kata seperti lowercase, uppercase, dan huruf berserta angka.
		<i>Tags</i>	Tipe data pada masukan tags tiket adalah <i>array</i> . Tags dapat berupa lebih dari satu kata. Selain itu juga dapat memiliki lebih dari satu tags dengan cara menggunakan pemisah koma.
		<i>Attachment</i>	<i>Attachment</i> pada API ini berupa file. File yang diterima oleh server berextensi <i>.csv</i> . Pada pengujian attachment dilakukan upload file dengan extension yang beragam seperti <i>.pdf</i> .
4.	<i>Customer Check Point Estimation</i>	<i>Product Name</i>	Nama produk pada API ini bertipe data <i>string</i> . Dalam pembuatan data uji dipilih beberapa kombinasi penulisan kata seperti lowercase, uppercase dan yang lainnya. Selain itu, dilakukan pengujian pada data <i>product</i> yang tidak ada pada database.
		<i>Quantity</i>	Jumlah pembelian dalam liter memiliki tipe data angka. Dalam pembuatan data uji dipilih beberapa angka yang termasuk dalam bilangan positif dan beberapa angka bilangan negative. Selain itu, juga dilakukan pengujian kombinasi angka dan huruf.
		<i>Price</i>	Harga produk yang akan dibeli memiliki tipe data berupa angka. Dalam pembuatan data uji dipilih beberapa angka yang termasuk dalam bilangan positif dan beberapa angka bilangan negative. Selain itu, juga dilakukan pengujian kombinasi angka dan huruf.
5.	<i>Get One dan Get URL Ticket Bulk Inject Point</i>	<i>ID</i>	Paramater ID memiliki tipe data berupa string. ID yang sesuai terdiri dari 16 digit dengan diawali 3 huruf BIP dan diikuti 13 digit angka. Dilakukan pengujian dalam keadaan jumlah digit yang tidak sesuai dan penulisan 3 huruf awal.

Pada penelitian ini, pengujian manual akan dibandingkan dengan pengujian otomatis menggunakan python. Pada pengujian manual, beberapa tahapan yang harus dilakukan antara lain:

1. Install Postman
2. Membuat *request* baru yang dapat dilakukan dengan menggunakan tambahan file biasa ataupun collection. Collection merupakan kumpulan dari beberapa file API.
3. Memilih *request method* yang tidak hanya terdiri dari method GET, POST, PUT, DELETE. *Method* dapat digunakan untuk menentukan jenis *request* terhadap API.
4. Mengisi komponen yang dibutuhkan seperti: parameter, authorization, body berfungsi untuk menyisipkan data.
5. Lakukan permintaan data (*request data*), jika semua data telah terisi yaitu method, endpoint, dan komponen seperti body.

Sedangkan pada pengujian otomatis, pytest digunakan sebagai kerangka kerja. Adapun

tahapan yang dilakukan pada pengujian otomatis antara lain adalah sebagai berikut.

1. Instalasi Python
2. Instalasi PIP
PIP adalah package untuk manajemen paket dalam Python. PIP digunakan untuk menginstall, menghapus, upgrade paket Python, dan lainnya. Pada penelitian ini, PIP digunakan untuk melakukan instalasi framework pytest. Instalasi PIP dilakukan ketika versi python yang digunakan dibawah 3.4.
3. Instalasi Pytest
Pytest digunakan untuk menjalankan script pengujian yang telah dibuat. Untuk menginstall pytest dapat menggunakan perintah “pip install pytest”.
4. Pembuatan *script* pengujian
Script pengujian dibuat menggunakan bahasa python. Untuk setiap skenario pengujian dibuat dalam satu file python. Setiap API yang saling berhubungan akan dikelompokkan dalam satu folder untuk mempermudah pengecekan apabila terdapat bug.
5. Menjalankan *script* pengujian
Langkah terakhir pada pengujian otomatis adalah menjalankan script yang telah dibuat. Pengujian dapat dilakukan pada terminal dengan beberapa cara yang bergantung pada file atau folder yang ingin diuji. Untuk menjalankan satu file tertentu maka dapat menggunakan perintah “\$ pytest test/nama_file.py”. Sedangkan apabila ingin menjalankan semua file yang ada dalam folder maka menggunakan perintah “\$ pytest tests/”. Hasil menjalankan script pengujian adalah muncul keterangan Pass atau Fail untuk setiap skenario pengujian pada terminal dan setiap data uji pada file excel hasil pengujian. Keterangan Pass digunakan apabila kode status pemanggilan API adalah 200 dan 201. Sedangkan keterangan Fail digunakan apabila kode status pemanggilan API adalah kelompok 400 dan 500.

4.3. Eksperimen Pengujian

Adapun tahapan yang dilakukan pada eksperimen pengujian adalah sebagai berikut.



Gambar 2. Tahapan dalam Eksperimen Pengujian

Pada pengujian manual menggunakan postman, setiap data yang akan diuji ditulis secara manual pada setiap masukan. Kemudian melakukan permintaan kepada REST API sesuai data yang telah dimasukkan. Setiap permintaan yang dilakukan akan mendapatkan hasil berupa data pengujian berhasil ataupun gagal. Data pengujian tersebut dapat disimpan dalam file berupa JSON atau text. Mekanisme pengujian otomatis dengan Python dapat ditunjukkan pada Gambar 3.



Gambar 3. Mekanisme Pengujian Otomatis dengan Python

Secara garis besar, proses permintaan pada REST API pada pengujian otomatis mirip dengan pengujian manual. Hal yang membedakan antara pengujian otomatis dan manual terletak pada bentuk masukan dan keluaran pengujian. Pada pengujian otomatis, semua data uji pada 1 API disimpan dalam sebuah file Excel. File tersebut akan dibaca oleh sistem dengan bantuan library openpyxl. Semua data yang tersimpan dalam file yang berhasil dibaca akan mengirimkan

permintaan pada REST API. Hasil dari permintaan yang telah dilakukan akan ditulis dan disimpan oleh sistem pada sebuah file excel dengan bantuan library yang sama yaitu openpyxl. Sehingga keluaran dari sistem adalah file excel yang menyimpan data hasil pengujian. Setiap API memiliki format file yang berbeda-beda sesuai dengan masukan setiap API.

Pengujian otomatis dilakukan berdasarkan skenario pengujian yang telah didefinisikan sebelumnya. Pengujian otomatis diimplementasikan menggunakan bahasa python dengan framework pytest. Setiap kasus uji akan diuji menggunakan 5 data uji, sehingga dalam setiap API diperlukan data uji yang lebih dari 10. Hal ini, akan menyulitkan apabila kita tetap memasukkan satu-persatu dalam kode. Oleh karena itu, dalam pengujian otomatis, data uji disimpan dalam file berekstensi excel. Data pada file tersebut digunakan sebagai masukan pada proses pengujian. Dan untuk hasil pengujian dari setiap data uji juga akan disimpan pada file excel yang sama. Dalam mengimplementasikan proses tersebut diperlukan bantuan library openpyxl yang digunakan untuk membuka dan menyimpan file excel. Gambar 4 menunjukkan kode pengujian otomatis untuk masing-masing API.

```
Kode Semu : API Testing
function test_API():
    work = openpyxl.load_workbook(file)
    for i in range(2, max_row+1):
        data = work.cell(row, column)
        response = request.method(url, headers, data)
        result = work.cell(row, column)
        if response.status_code == 200:
            result.value = "PASS"
        else:
            result.value = "FAIL"
    work.save(file)
end function
```

Gambar 4. Potongan Gambar Kode Semu API Testing

Salah satu cara yang dapat digunakan untuk mengakses HTTP dalam bahasa python adalah menggunakan library request. Hal yang perlu dilakukan adalah memanggil fungsi request.methodname (params). Nama method dapat diisi get, delete, post, ataupun method yang lain. Sedangkan params dapat diisi dengan url, data dan kebutuhan lainnya. Library lain yang digunakan pada pembuatan pengujian otomatis ini adalah openpyxl. Library ini berguna untuk mengakses file berekstensi xlsx. Untuk membuka file excel dapat menggunakan fungsi load_workbook dan save untuk menyimpan serta menutup file excel. Selain itu, untuk mendapatkan sheet yang sedang berjalan dapat menggunakan fungsi active.

4.4. Hasil Eksperimen Pengujian

Hasil pengujian didapatkan setelah melakukan pengujian manual menggunakan postman dan pengujian otomatis menggunakan python serta pytest pada 7 API modul loyalti aplikasi MyPertamina. Terdapat total 30 test case skenario pengujian dan 155 data yang diujikan. Hasil yang didapat yakni seluruh kasus uji yang diujikan sesuai dengan yang diharapkan. Hal ini menunjukkan bahwa 7 API modul loyalti aplikasi MyPertamina sudah berhasil memenuhi lingkup pengujian yang telah dibuat. Pengujian manual dilakukan untuk menentukan akurasi dari implementasi pengujian otomatis. Dari hasil yang diperoleh, pengujian otomatis memiliki tingkat akurasi sama dengan pengujian manual. Sehingga dapat disimpulkan bahwa pengujian secara otomatis memiliki akurasi 100%.

Dari pengujian otomatis yang sudah dilakukan dapat ditentukan poin-poin keunggulan yang didapat dari sisi efisiensi. Poin pertama yakni mengenai waktu berjalannya pengujian otomatis. Pengujian otomatis menghabiskan waktu 52 detik. Waktu ini digunakan untuk melakukan sebanyak 7 API dengan 155 data uji. Waktu pengujian tercepat untuk 1 API hanya

3.98 detik saja. Waktu pengujian terlama adalah 16.76 detik. Sedangkan rata-rata secara keseluruhan waktu pengujian yang dijalankan adalah 7.44 detik. Jika diasumsikan rata-rata waktu pengujian yang didapat untuk pengujian secara manual kurang lebih 15 detik untuk 1 data uji, maka total waktu yang dibutuhkan untuk 155 data uji adalah 38 menit 45 detik. Sehingga dengan pengujian otomatis kita dapat menghemat waktu selama 37 menit 53 detik.

Setiap API memiliki waktu pengujian yang berbeda-beda. Perbedaan ini muncul sesuai dengan perbedaan jumlah test case yang ada dalam skenario pengujian. Skenario dengan test case dan data masukan yang lebih banyak akan berjalan lebih lama dibandingkan dengan skenario yang testcase dan data masukannya lebih sedikit. Selain itu, lama pengujian juga dipengaruhi oleh kompleksitas sistem dan koneksi yang digunakan untuk pengujian. Koneksi internet yang lambat pastinya akan memperlambat proses pengujian yang dijalankan. Hal ini karena, pengujian terhadap HTTP request memerlukan koneksi internet untuk menjalankan.

Keunggulan kedua adalah hasil pengujian otomatis disimpan dalam satu file Excel. Dalam file ini tersimpan seluruh data uji dari masing-masing API dan terdapat pesan error yang ada. Dengan terkumpulnya menjadi satu, memudahkan penguji ataupun pengembang untuk mencari dan mencoba kembali data uji yang bermasalah. Untuk sisi efektivitas, kedua bentuk pengujian efektif untuk dilakukan. Hal ini karena hasil kedua pengujian didapatkan data kesesuaian bahwa ketujuh API mendapatkan 100% hasil yang sama dari pengujian manual dan otomatis. Dari hasil kesesuaian tersebut, diketahui bahwa pengujian otomatis telah terimplementasi dengan baik dan sesuai harapan. Dari kedua hasil tersebut, menunjukkan bahwa semua API berhasil memenuhi kebutuhan pengguna atau telah sesuai skenario pengujian dan dapat digunakan dengan baik.

5. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diperoleh dari penelitian ini antara lain adalah sebagai berikut.

1. Pengujian diawali dengan analisis kebutuhan berdasarkan buku panduan, dilanjutkan dengan pengembangan skenario pengujian dan pembuatan data uji sesuai dengan Teknik *equivalence partitioning* tanpa mengetahui kode program.
2. Penerapan teknik *equivalence partitioning* dilakukan dengan membagi ruang pilihan atau kemungkinan yang ada baik pada skenario pengujian maupun data uji dan berfokus pada pengujian masukan sistem.
3. Hasil pengujian manual dan otomatis RESTful API aplikasi MyPertamina modul loyalti menunjukkan bahwa ketujuh API telah mendapatkan hasil sesuai dengan yang diharapkan dan dapat berjalan dengan baik.
4. Pengujian otomatis lebih efisien dilakukan daripada pengujian manual berdasarkan total waktu yang dibutuhkan untuk menguji ketujuh API yaitu selama 52.12 detik atau hampir 1 menit.

Saran dalam penelitian ini antara lain adalah sebagai berikut.

1. Pengujian otomatis yang telah dibuat perlu disederhanakan lagi agar memberikan hasil yang lebih efisien dan membutuhkan waktu yang sedikit dalam pengujiannya.
2. Perlu menambahkan beberapa kondisi atau skenario yang belum dapat ditelusuri secara otomatis sebelumnya.

DAFTAR PUSTAKA

- [1] M. Prawiro, "Pengertian Smartphone, Sistem Operasi, Fitur, dan Jenis Smartphone," Mei 2018. [Online]. Available: <https://www.maxmanroe.com/vid/teknologi/mobile-app/pengertian-smartphone.html>. [Accessed Mei 2022].
- [2] L. Ceci, "Number of apps available in leading app stores as of 1st quarter 2022," April 2022. [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>. [Accessed Mei 2022].

- [3] A. Lawrence, "API: Pengertian, Fungsi, dan Cara Kerjanya," Niagahoster, October 2020. [Online]. Available: <https://www.niagahoster.co.id/blog/api-adalah/>. [Accessed Mei 2022].
- [4] P. B. A., A. D. B., S. M., A. N. T. S. and S. A., "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 3, p. 150, 2020.
- [5] N. S., "Black Box and White Box Testing Techniques – A Literature Review," *International Journal Embeded System Application*, vol. 2, no. 2, pp. 29-50, 2012.
- [6] L. Siagian, *Otomatisasi Pengujian Perangkat Lunak (Software Test Automation)*, Yogyakarta: Deepublish, 2018.
- [7] S. Bansal, "Rest API (Introduction)," *Software Testing Help*, May 2022. [Online]. Available: <https://www.softwaretestinghelp.com/rest-api-response-codes/>. [Accessed 2023].
- [8] S. I. A., P. J., P. P., S. R. and S. A., "Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalence Partitions,," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 4, no. 1, pp. 38-45, 2021.
- [9] Fathurrahman, "Apa itu Postman dan Bagaimana menggunakan Postman untuk Menguji API?," *Matawebsite*, November 2021. [Online]. Available: <https://www.matawebsite.com/blog/apa-itu-postman-dan-bagaimana-menggunakan-postman-untuk-menguji-api>. [Accessed Mei 2023].
- [10] . F. Choliludin, "Pytest: Memahami Cara Kerja Testing Framework (bagian 1)," Mei 2022. [Online]. Available: <https://fachrul.id/belajar-pytest-framework-1/>. [Accessed 2023].
- [11] A. Prasetyo, *Implementasi Automated Functional Ui Testing Untuk Mencapai Konsep CI/CD Terintegrasi Jenkins Menggunakan Framework Serenity Dengan Page Object Model Design Pattern Pada Ppdb Jatim*. Undergraduate thesis, Institut Teknologi Sepuluh Nopember. [thumbnail of, Surabaya: ITS Repository, 2021.
- [12] I. A. Shaleh, J. Prayogi, P. Pirdaus, R. Syawal and A. Saifudin, "Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 4, no. 1, pp. 38-45, 2021.
- [13] M. Iskhak and S. Rizkika, "Implementasi metode pengujian equivalence partitioning pada pengembangan RESTful API Sistem Informasi Klinik Pratama UPN "Veteran" Yogyakarta," in *Seminar Nasional Informatika*, Yogyakarta, 2022.
- [14] M. Sholeh, I. Gisfas, Cahiman and M. A. Fauz, "Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods," *Open Access proceedings Journal of Physics: Conference series*, pp. 1-8, 2020.
- [15] G. W. Sasmito and M. A. Mutasodirin, "Black Box Testing with Equivalence Partitions Techniques in Transcrop Applications," in *International Conference of Computer and Informatics Engineering (IC2IE)*, Lombok, 2023.
- [16] R. D. Siahaan, S. S. Kusumawardani and I. Hidayah, "E-Learning Evaluation of Del Superior High School Based on Black Box Testing with Equivalence Partitioning and Graph-Based Testing," in *International Conference on Science and Technology (ICST)*, Yogyakarta, 2023.