

Perancangan Automated Testing Pada Studi Kasus Website Indicar

Bima Prakoso^{1*}, Ari Sujarwo²

^{1,2}Jurusan Informatika, Universitas Islam Indonesia

¹19523151@students.uii.ac.id

²ari.sujarwo@uui.ac.id

*Corresponding author

Abstrak— *Internet of Things (IoT)* adalah sebuah perangkat elektronik yang terhubung dengan website yang mengontrol / memonitornya. Dalam memastikan website tersebut sudah terhubung dengan perangkat IoT, perlu adanya pengujian baik itu secara manual ataupun *automated*. Pada penerapan pengujian secara manual memerlukan waktu dan tenaga manusia yang lebih banyak untuk menguji website. Sedangkan *automated testing* dapat menyelesaikan permasalahan tersebut. Tools yang digunakan dalam proses pengujian secara *automated* adalah Cypress. Cypress memiliki definisi dimana sebuah *framework* pengujian yang berguna untuk memudahkan para developer dalam melakukan pengujian secara berkala. Lalu, Cypress juga dapat membantu developer melakukan pengecekan fitur website apabila ada perubahan fitur yang ada di server maupun migrasi server. Untuk itu, penggunaan *framework* Cypress akan membantu dalam pembuatan *script automated testing* yang ada pada website Indicar Admin dan Indicar Webapp.

Kata Kunci— Pengujian IoT, Pengujian manual, *Automated testing*, Cypress, Indicar.

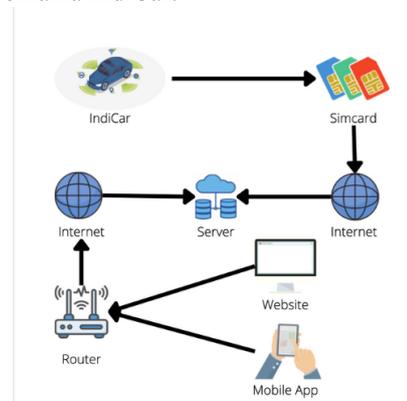
I. PENDAHULUAN

Seiring perkembangan teknologi yang ada, banyak perusahaan atau instansi yang menjadikan teknologi sebagai solusi dari permasalahan yang ada. Salah satu contohnya adalah perangkat *internet of things (IoT)*. *Internet of Things (IoT)* adalah sebuah perangkat elektronik yang dapat berkomunikasi dengan pengguna yang berfungsi untuk memonitor atau mengontrol perangkat lain melalui internet [1]. Dengan adanya perangkat IoT, kita akan lebih mudah untuk memonitor ataupun melakukan sebuah perintah kepada benda yang dipasangkan oleh perangkat IoT seperti kendaraan, gagang pintu, dan masih banyak lagi.

Untuk memastikan bahwa aplikasi siap dipasarkan, perlu adanya pengujian yang benar. Hal ini sangat penting untuk menyakinkan pelanggan bahwa aplikasi sudah layak dan telah memenuhi persyaratan yang ada [2]. Dengan melakukan pengujian juga, perusahaan dapat mengantisipasi kerugian yang ada di masa depan [3]. Apabila perusahaan memaksakan untuk memasarkan aplikasi yang belum lolos dalam tahap pengujian, maka salah satu dampaknya adalah membuat nama perusahaan tersebut menjadi buruk. Dari sisi pelanggan sendiri akan menjadi tidak nyaman untuk menggunakan aplikasi tersebut dan berpindah ke perusahaan lainnya yang memiliki aplikasi sejenis.

Oleh karena itu, perlu adanya pengujian aplikasi user interface atau bisa disebut *front-end testing* [4]. *Front-end*

testing yang berfungsi untuk melihat apakah fitur yang dibuat sudah terhubung ke perangkat IoT atau belum. Pengujian *front-end testing* yang dilakukan secara manual atau secara *automated*. Untuk pengujian secara manual terhadap semua fitur memiliki banyak kekurangan seperti membutuhkan waktu yang banyak, memerlukan orang yang cukup, dan masih banyak lagi [5]. Maka dari itu, untuk menutupi semua kekurangan tadi diperlukan suatu pengujian yang dilakukan secara *automated* yang dapat memudahkan kita dalam pengujian fitur secara berulang dengan membutuhkan waktu yang lebih sedikit. Untuk itu pada studi kasus pengujian aplikasi *internet of things (IoT)* berbasis website yang dikembangkan oleh Direktorat Digital Bisnis (DDB) Telkom Indonesia bernama Indicar.



Gbr. 1 Skema Indicar

Untuk Indicar sendiri memiliki skema dimana perangkat IoT tersebut memiliki kartu yang mempunyai paket internet dan telah terpasang di kendaraan. Perangkat tersebut akan mengirimkan data realtime kendaraan melalui internet ke dalam server Indicar. Data yang disimpan di dalam server dan apabila ingin diakses dapat melalui website Indicar.

Indicar ini sendiri terbagi menjadi dua jenis website yang berguna memudahkan dalam mengelolanya. Website yang pertama adalah website yang digunakan untuk staff Indicar itu sendiri dan website kedua digunakan oleh pelanggan bernama webapp. Selain menggunakan website, Indicar juga menyediakan aplikasi yang dapat di download melalui Google Play Store khusus untuk para pelanggannya.

Untuk pengujian fitur yang ada di website yang dilakukan secara *automated* memiliki beragam *tools* seperti Selenium webdriver, Katalon Studio, testIM, Cypress, dan masih banyak lagi. Dari beragam *tools* yang ada, saya memilih Cypress sebagai pengujian secara *automated*. Alasan saya

dalam menggunakan Cypress adalah karena saran dari supervisor saya serta tim developer yang ada di DDB Telkom. Tidak hanya itu, Cypress juga memiliki banyak *command* di dalamnya, menghadirkan dashboard yang iterative bernama “Test runner” yang berguna untuk memvisualisasikan pengujian secara bertahap dan sangat lengkap, mudah digunakan, gratis, serta memiliki komunitas yang luas yang berasal dari tim pengembang dan tim penguji yang professional [6]. Untuk pengaplikasian dari Cypress ini sendiri, kita sebagai pengguna harus membuat script yang ada, melihat apakah ada yang error atau tidak dan apabila tidak ada yang error, maka dapat dipresentasikan kepada para developer serta supervisor. Dari sanalah, para developer akan sangat terbantu untuk mengecek fitur berjalan dengan normal atau ada masalah saat ada perubahan pada server.

II. METODOLOGI

Untuk mempermudah melakukan perbandingan, penulis telah merangkum beberapa point perbandingan terhadap Cypress. *Tools* yang pertama ada Selenium webdriver, Berdasarkan pengujian yang ada menyebutkan bahwa selenium webdriver adalah sebuah *tools* pengujian *automated* gratis yang dapat berkomunikasi dengan banyak browser dan dapat digunakan untuk aplikasi Ajax [7]. Selenium webdriver memiliki banyak *command* di dalamnya dan memiliki komunitas yang luas. Namun dari banyaknya kelebihan yang ada, terdapat kekurangan dari Selenium webdriver sendiri seperti tidak adanya *dashboard* yang memvisualisasikan pengujian secara bertahap.

Untuk *tools* berikutnya ada Katalon Studio, Katalon Studio adalah sebuah *tools* pengujian *automated* gratis yang menggunakan bahasa Java dan mendukung pengujian web, desktop hingga seluler. Cara kerja Katalon Studio adalah dengan cara merekam Langkah pengujian dan *scripting* untuk menjalankan pengujian [8]. Katalon Studio sendiri pertama kali rilis tahun 2015 sampai saat ini sudah memiliki komunitas yang cukup besar. Akan tetapi, dari semua kelebihan yang ada katalon memiliki kode yang lebih sulit untuk dipelajari.

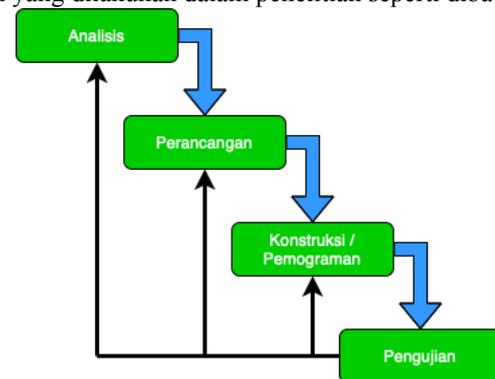
Dan perbandingan *tools* yang terakhir adalah testIM. Menurut jurnal yang berjudul “*AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW*” testIM adalah sebuah *tools* pengujian *automated* interface berbasis website. Cara kerja dari testIM sendiri hampir mirip dengan Katalon Studio. Dengan adanya interface berbasis website, akan lebih mudah dalam pengerjaan. testIM sendiri memiliki fitur untuk pengujian data, menghubungkan CSV, Excel, ataupun file JSON [9]. Sayangnya, dari segala kemudahan yang diberikan testIM ini tidaklah sepenuhnya gratis karena dalam satu bulan kita hanya diberikan gratis menjalankan pengujian sebanyak 1000 kali. Selain itu, testIM masih tergolong baru sehingga sulitnya komunitas yang besar dan professional. Selain itu testIM masih memiliki fitur yang sedikit apabila dibandingkan dengan para pesaingnya.

TABEL I
PERBANDINGAN CYPRESS DENGAN *AUTOMATION TOOLS* SERUPA

Perbandingan	Automation Tools			
	Cypress	Selenium webdriver	Katalon Studio	testIM
Dahboard yang iteratif	?	X	?	?
Mudah digunakan	?	?	X	?
Gratis	?	?	?	X
Komunitas yang luas	?	?	?	X
Fitur yang lengkap	?	?	?	X

III. METODOLOGI

Tahapan yang penulis gunakan adalah metode *Research and Development* (R&D). Metode ini digunakan untuk mendapatkan produk yang efektif [10]. Untuk tahapan-tahapan yang dilakukan dalam penelitian seperti dibawah ini.



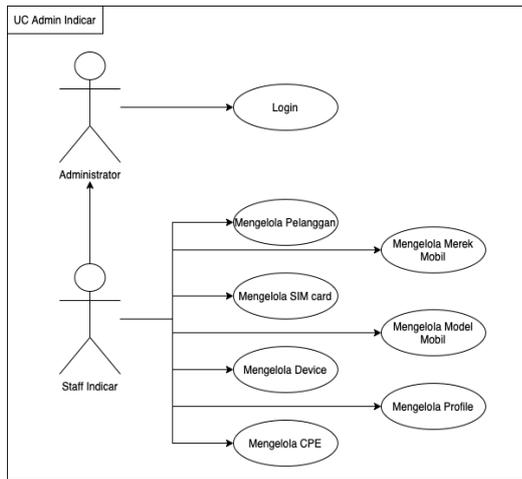
Gbr. 2 Tahapan merancang pengujian

A. Analisis

Pada tahapan analisis ini, penulis mengumpulkan data primer dari pihak Telkom dan melakukan wawancara kepada tim developer. Selain itu, penulis mengumpulkan data sekunder melalui studi kepustakaan, artikel, dan jurnal yang ada di internet.

B. Perancangan

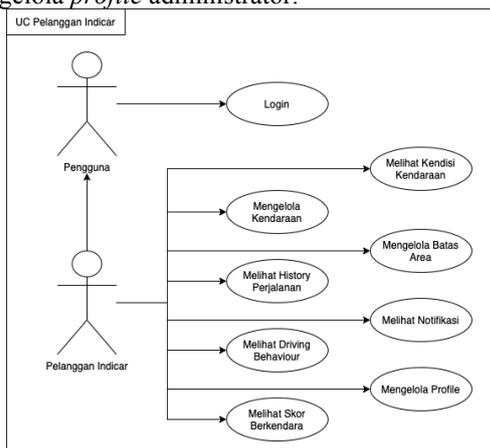
Pada tahapan ini, penulis mengumpulkan masalah yang didapatkan dari tim developer. Masalah-masalah tersebut dirangkum menjadi dua hak akses.



Gbr. 3 Use Case Administrator

Pada halaman administrator, terdapat fitur seperti:

- Melakukan *login*,
- Mengelola pelanggan,
- Mengelola *SIM card* pelanggan yang digunakan pelanggan IndiCar,
- Mengelola *device Internet of Things (IoT)* yang dipasang pada kendaraan pelanggan IndiCar,
- Mengelola *CPE (Costumer Premises Equipment)* pelanggan IndiCar,
- Mengelola *manufacture / merek* kendaraan yang digunakan pelanggan Indicar,
- Mengelola model kendaraan yang digunakan oleh pelanggan IndiCar,
- Mengelola *profile* administrator.



Gbr. 4 Use Case Pelanggan IndiCar

Pada halaman pelanggan IndiCar, terdapat fitur seperti:

- Melakukan *login*,
- Mengelola kendaraan yang dipasangkan perangkat *Internet of Things (IoT)*,
- Melihat history perjalanan,
- Melihat *driving behaviour / kebiasaan* pelanggan saat sedang mengendarai kendaraan,
- Melihat skor berkendara,

- Melihat kondisi kendaraan yang dipasangkan perangkat *Internet of Things (IoT)*,
- Mengelola batas area yang ada pada kendaraan,
- Melihat notifikasi,
- Mengelola *profile*.

C. Konstruksi / Pemograman

Pada tahapan ini penulis melakukan pemograman NodeJS yang dipasangkan tools *Cypress* untuk menjalankannya. Tidak hanya itu, penulis juga menggunakan VS Code sebagai alat untuk menulis kode pemograman. Selain itu, penulis juga menggunakan Electron 94 dimana hal tersebut berfungsi untuk menjalankan *dashboard* Cypress. Elektron 94 merupakan browser bawaan yang ada saat melakukan instalasi Cypress pada *project*.

D. Pengujian

Tahapan ini merupakan tahapan pengujian kode pemograman yang telah dibuat apakah kode program yang dibuat telah sesuai dengan perancangan pengujian yang telah dibuat developer apa belum, apakah ada yang error atau tidak. Apabila kesalahan pada proses konstruksi / pemograman, maka perlu kembali ke tahapan konstruksi / pemograman lagi. Apabila tidak ada yang error dan sesuai maka akan dipresentasikan kepada supervisor dan tim developer agar dapat berguna kedepannya.

IV. HASIL DAN PEMBAHASAN

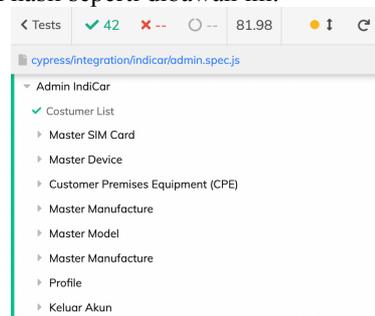
Pada bagian ini, penulis akan membandingkan hasil antara pengujian secara manual dan pengujian *automated*.

A. Proses pengujian manual di website IndiCar

Untuk pengujian yang dilakukan pada saat proses migrasi server pada tanggal 10 febuari 2022, saya yang dibantu developer saat proses pengujian fitur setelah proses migrasi menghasilkan total waktu 30 menit. Dimana setiap website Indicar serta listener dilakukan pengujian secara bersamaan, dan setiap orang menghasilkan waktu kurang lebih 10 menit. Sehingga perlu adanya pengujian website secara *automated*.

B. Proses pengujian di website admin IndiCar

Untuk pengujian pada website admin Indicar, mendapatkan hasil seperti dibawah ini.



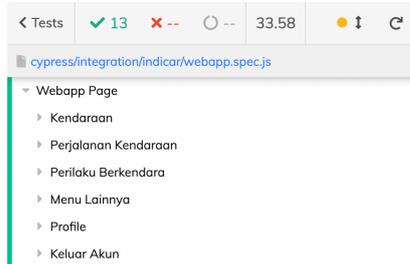
Gbr. 5 Proses pengujian automated website admin IndiCar

Pada proses pengujian dengan menggunakan Cypress pada website admin IndiCar, didapatkan waktu 82 detik dengan 42 pengujian telah berhasil melewati test uji. Pengujian tersebut meliputi melihat *customer*, CRUD (*create, read, update, delete*) *simcard*, CRUD *device* (perangkat IoT), CRUD

Customer Premises Equipment (CPE), *CRU manufacture* (nama perusahaan kendaraan), *CRUD model* (model kendaraan), hapus *manufacture*, melihat dan mengubah *profile*, dan keluar akun. Hal ini sangat menghemat waktu para developer untuk mengecek secara berkala.

C. Proses pengujian di website pelanggan IndiCar

Pada pengujian website webapp IndiCar, mendapatkan hasil seperti dibawah ini.



Gbr. 6 Proses pengujian *automated* website admin IndiCar

Pada proses pengujian dengan menggunakan Cypress pada website pelanggan IndiCar, didapatkan waktu 33 detik dengan 13 pengujian telah berhasil melewati test uji. Pengujian tersebut meliputi melihat kendaraan, melihat *history* perjalanan kendaraan, melihat perilaku pengemudi selama berkendara, melihat menu lainnya (skor berkendara, kondisi kendaraan, batas area, dan laporan gangguan), melihat dan mengubah *profile*, dan keluar akun. Sama seperti website admin IndiCar, pengujian ini sangat menghemat waktu saat melakukan pengujian secara berkala.

V. KESIMPULAN

Dari bab hasil dan pembahasan, dapat dibuktikan bahwa permasalahan yang diangkat telah diselesaikan. Hasil yang saya dapatkan setelah membandingkan antara pengujian manual dan *automated* dapat diketahui bahwa pengujian dengan menggunakan Cypress lebih efisien dari segi waktu dan juga jumlah orang. Dengan adanya pengujian menggunakan Cypress, waktu pengujian yang awalnya 30 menit menjadi 82 detik. Tidak hanya itu, pengujian menggunakan Cypress juga menghemat tenaga manusia yang awalnya tiga orang menjadi satu orang. Dalam penerapannya, Cypress juga dapat membantu para developer untuk menguji sistem secara berkala apabila ada perubahan fitur yang ada di server ataupun migrasi server.

REFERENSI

- [1] A. Surahman, B. Aditama, M. Bakri and Rasna, "SISTEM PAKAN AYAM OTOMATIS BERBASIS INTERNET OF THINGS", 2021.
- [2] A. Purnomo, "Software Testing Aplikasi Website PT Gramedia Menggunakan Metode Blackbox pada PT WGS Bandung", 2013.
- [3] Arfinda D., Selo S., Lukito L., "Studi Literatur Pengujian Perangkat Lunak", 2019.
- [4] Subroto B., "Back-End SQL Express dan *Front-End Access*: Membuat Aplikasi Dokumen Elektronik Sederhana", 2020.
- [5] Z. Fatimatuz. P. Fajar, A. Achmad, "Kakas Bantu untuk Penentuan Prioritas *Test Scenario* Berdasarkan *UML Activity Diagram*". 2019.
- [6] T. Malika Tasnim, "AUTOMATED WITH CYPRESS", 2021.
- [7] G. Satish, J. Rahul, G. Dhanashree, "Analysis and Design of Selenium WebDriver Automation Testing Framework", 2015.

- [8] W. Ieda Rosana K., R. André Fernando, A. Joseane, S. Maikon C. D., D. Ricardo V., M. Vanessa L., "BENEFÍCIOS DA AUTOMAÇÃO DE TESTES COM O USO DA FERRAMENTA KATALON STUDIO", 2021.
- [9] A. Dilara, M. Alok, "AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW", 2020.
- [10] Sugiyono, "Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R&D, Alfabeta", Bandung, Cet ke-19, 2014.