

BAB III

TINJAUAN PUSTAKA

3.1 Software Testing

Merupakan prosedur dalam mengeksekusi program atau aplikasi dengan tujuan untuk mencari adanya kesalahan atau error. Proses ini memvalidasi dan memverifikasi seluruh komponen dari sistem dan memeriksa hasil sesuai dengan yang diharapkan [1]

Beberapa tingkatan pada pengujian sebuah aplikasi atau software :

1. Unit testing, merupakan pengujian pada level terendah. Pengujian ini menguji dasar unit dari software, atau disebut juga sebagai module testing. Pengujian ini memverifikasi fungsionalitas dari kode fungsi, *object-oriented*, atau level *class*.
2. Integration Testing, merupakan pengujian pada unit yang lebih besar pada segi strukturalnya. Pengujian ini dilakukan antara komponen stuktur yang dibuat. Sehingga pengujian integrasi berfokus pada tipe dari aplikasi yang memverifikasi antarmuka antara komponen dengan rancangan aplikasi. Pengujian ini berguna untuk menemukan *error* pada antarmuka dan interaksi antara komponen didalamnya.
3. System Testing, merupakan pengujian *end-to-end* dari sistem. Pengujian sistem yang akan dilakukan berdasarkan fungsi dari spesifikasi sistem, dapat juga berupa Non-functional yang terdiri dari keamanan, *reliability*, *maintainablity*
4. Acceptance Testing, merupakan persiapan dari produk atau aplikasi. Pada bagian ini, pengujian fokus pada operasional atas persiapan sistem yang sudah selesai dari pengembang kepada pengguna. tujuan pengujian ini adalah untuk meyakinkan bahwa aplikasi sudah dapat berjalan
5. Alpha Testing, merupakan tahap dimana pengujian dilakukan oleh pengguna atau pelanggan

6. Beta Testing, merupakan pengujian yang dilakukan oleh pengguna khusus dimana mereka dapat menguji dan memeriksa bahwa produk dapat berjalan dan memiliki sedikit kesalahan
7. Regression Testing, merupakan fokus pengujian pada pencarian *error* ketika adanya perubahan kode. Biasanya pengujian ini berfokus pada *bug* lama yang muncul kembali. Metode umum dari pengujian ini adalah menjalankan kembali pengujian yang sudah dilakukan sebelumnya dikarenakan adanya penambahan fitur tertentu, atau perubahan.

Pada proses kerja magang ini, penulis melakukan pengujian fungsional pada aplikasi *react* yang akan dilakukan secara otomatis. Parameter yang akan diperhatikan pada implementasi pengujian ini adalah:

1. Keberhasilan pengujian (*tombol, search form*)
2. Pemeriksaan list yang masuk saat melakukan proses *create*
3. Waktu pengujian

3.2 WebdriverIO

WebdriverIO merupakan *testing tools open source utility* Node.js. sebuah testing tools yang sudah disederhanakan dibanding dengan Selenium. Test ini dilakukan menggunakan bahasa pemrograman Javascript. WebdriverIO merupakan *framework* automasi yang dibangun untuk mengotomasikan web dan aplikasi mobile[2]. WebdriverIO menyederhanakan interaksi dengan aplikasi yang dibuat dengan menyediakan plugin yang membantu pengguna untuk membuat pengujian yang *salable, robust, dan flakiness*. WebdriverIO dirancang untuk:

- a. *Extendable*. Fitur bantuan, kombinasi command yang simpel dan membantu
- b. *Compatible*. Dapat dijalankan pada *cross-browser testing*
- c. *Feature Rich*. Banyak variasi *built-in* dan *plugins* yang membantu pengguna untuk mengintegrasikan dan menyebarkan perancangan sesuai dengan kebutuhan

3.3 React.js

React adalah library Javascript terpopuler untuk membuat user interface (UI). Tool ini menawarkan respons cepat untuk user input dengan menggunakan metode baru dalam proses rendering website.[3]

Komponen dari tool ini dikembangkan oleh Facebook. Dahulunya tool ini diluncurkan sebagai tool open-source JavaScript di tahun 2013. Saat ini, React tetap terdepan mendahului kompetitornya seperti Angular dan Bootstrap, dua library JavaScript terlaris.

Commsult Indonesia menggunakan *react.js* sebagai salah satu pemrograman untuk aplikasi mereka.

3.4 Komponen Pengujian

Pada poin ini, penulis akan menjelaskan mengenai komponen apa saja yang diperlukan untuk menunjang implementasi sistem.

a. Sistem Operasi

Penulis menggunakan laptop dengan deskripsi sebagai berikut

1. Laptop HP Pavilion AB-032AX
2. Sistem Operasi Windows 8.1 64 bit
3. RAM 8 GB
4. Prosesor AMD A10 – 8700P
5. GPU Radeon Dual Graphic R6&R7

b. Node.js merupakan sebuah *platform* yang digunakan untuk menjalankan kode JavaScript pada server.

Penulis menggunakan spesifikasi node.js dengan versi 14.11.0

```
PROBLEMS OUTPUT DEB
Filbertw@filbert MINGW
$ node -v
v14.11.0
```

Gambar 3.1 Versi Node.js

- c. NPM merupakan Node Package Manager yang digunakan untuk menginstal modul – modul , dan mengelola dependensi.

Penulis menggunakan NPM dengan versi 6.14.8

```
Filbertw@filbert MINGW
$ npm -v
6.14.8
```

Gambar 3.2 Versi npm

- d. Visual Studio Code

Untuk *code editor* , penulis menggunakan Visual Studio Code sebagai media untuk menuliskan kode. Penulis akan menggunakan bahasa pemrograman JavaScript

- e. Command Prompt

Penulis menggunakan Node.js Command Prompt untuk menjalankan aplikasi React dan Menjalankan proses pengujian

- f. WebdriverIO

WebdriverIO merupakan sebuah *tools open source* yang digunakan untuk melakukan sebuah pengujian aplikasi khususnya pada antarmuka sebuah browser. WebdriverIO akan penulis gunakan untuk menguji aplikasi ini.

- g. Browser (Google Chrome)

Browser digunakan untuk menjalankan aplikasi web secara otomatis. Browser yang penulis gunakan adalah Google Chrome dengan Versi 88.0.4324.150



Gambar 3.3 Versi Google Chrome

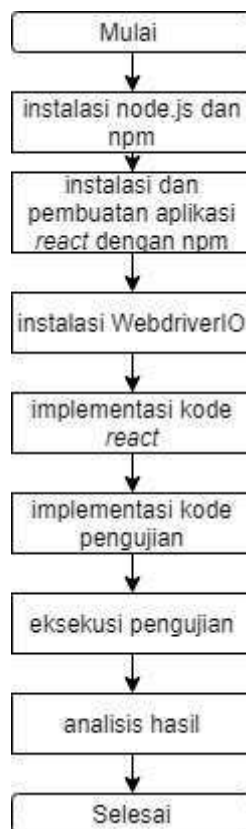
h. Chromedriver

Merupakan *webdriver open source* yang digunakan untuk melakukan proses pengujian secara otomatis

3.5 Alur Implementasi

Pada tahap ini, penulis akan menjelaskan mengenai alur implementasi dengan menggunakan *flowchart* guna mempermudah proses implementasi.

3.5.1 *Flowchart* implementasi



Gambar 3.4 Alur Implementasi

3.5.2 Tabel Skenario Uji

Tabel 3.1 Skenario Uji

No.	Skenario Uji	Deskripsi Pengujian
1	Open browser	Membuka browser
2	Enter name 1	Memasukkan nama pertama
3	update	Update list pertama
4	Get value and id check	Memeriksa nilai dan id list yang masuk
5	Create name 2	Memasukkan nama kedua
6	delete	Menghapus nama di list pertama
7	Enter name in loop	Memasukkan nama dalam loop
8	Enter name in loop	Memasukkan nama dalam loop
9	Enter name in loop	Memasukkan nama dalam loop

3.6 Proses Instalasi

Pada poin ini, penulis akan menjelaskan mengenai tahapan instalasi *software* atau *tools* yang diperlukan untuk melakukan pengujian

3.6.1 Instalasi Node.js dan NPM.

Proses instalasi Node dan NPM akan sekaligus menjadi satu bagian

- a. Download Node.js installer pada laman <https://nodejs.org/download/>[4]



Gambar 3.5 Halaman Instalasi Node.js

- b. Berdasarkan pada spesifikasi yang penulis pakai, penulis menggunakan Windows Installer(.msi) 64 bit
- c. Setelah selesai, maka akan ditampilkan proses instalasi



Gambar 3.6 *Setup* Node.js

- d. Setelah selesai, maka node.js dan NPM sudah terinstall

3.6.2 Instalasi React menggunakan NPM

Pada tahap ini, penulis akan menjelaskan tentang tahapan untuk menginstal dan membuat aplikasi react

- a. Membuat aplikasi baru dengan command npx

```
C:\Users\Filbertw>
C:\Users\Filbertw>
C:\Users\Filbertw>npm create-react-app basic-crud-react-master
```

Gambar 3.7 Pembuatan aplikasi *react*

- b. Masuk ke direktori

```
C:\Users\Filbertw>cd basic-crud-react-master
C:\Users\Filbertw\basic-crud-react-master>
```

Gambar 3.8 Masuk dalam direktori

- c. Jalankan aplikasi dengan NPM start. Aplikasi sudah dapat berjalan

```
npm
Your environment has been set up for using Node.js 14.11.0 (x64) and npm.
C:\Users\Filbertw>cd basic-crud-react-master
C:\Users\Filbertw\basic-crud-react-master>
C:\Users\Filbertw\basic-crud-react-master>
C:\Users\Filbertw\basic-crud-react-master>npm run start
> basic-crud@0.1.0 start C:\Users\Filbertw\basic-crud-react-master
> react-scripts start

Compiled successfully!

You can now view basic-crud in the browser.

  Local:            http://localhost:3000/
  On Your Network: http://192.168.1.6:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Gambar 3.9 Menjalankan npm start

3.6.3 Instalasi WebdriverIO

Untuk menginstal WebdriverIO dibutuhkan Node.js dengan versi minimum 12.16.1

- a. Menjalankan command npm install @wdio/cli pada terminal

```
C:\Users\Filbertw>npm install @wdio/cli
```


Gambar 3.10 Menjalankan instalasi WebdriverIO

- b. Menjalankan command `npx wdio config`. Dikarenakan proyek akan dilakukan pada lokal, maka pilih *local machine*

```
C:\Users\Filbertw\webdriverio-test>npx wdio config
=====
WDIO Configuration Helper
=====
? Where is your automation backend located? <Use arrow keys>
? Where is your automation backend located?
? Where is your automation backend located?
> On my local machine
  In the cloud using Experitest
  In the cloud using Sauce Labs
  In the cloud using Browserstack or Testingbot or LambdaTest or a different service
  I have my own Selenium cloud
```

Gambar 3.11 Pemilihan tempat pengujian

- c. Selanjutnya pilih *mocha* untuk framework, karena framework yang digunakan adalah *mocha*

```
? Where is your automation backend located? On my local machine
? Which framework do you want to use? <Use arrow keys>
> mocha
  jasmine
  cucumber
```

Gambar 3.12 Konfigurasi WebdriverIO

- d. Penulis menggunakan command asinkronus untuk menjalankan WebdriverIO

```
? Which framework do you want to use? mocha
? Do you want to run WebdriverIO commands synchronous or asynchronous? <Use arrow keys>
? Do you want to run WebdriverIO commands synchronous or asynchronous?
  sync
> async
```

Gambar 3.13 Konfigurasi asinkronus

- e. Untuk reporter, penulis menggunakan *spec reporter*

```
Do you want to run WebDriverIO commands synchronous or asynchronous? async
? Where are your test specs located? ./test/specs/**/*.js
? Do you want WebDriverIO to autogenerate some test files? Yes
? Do you want to use page objects (https://martinfowler.com/bliki/PageObject.htm
? Do you want to use page objects (https://martinfowler.com/bliki/PageObject.htm
? Do you want to use page objects (https://martinfowler.com/bliki/PageObject.htm
1)? No
? Are you using a compiler? No!
? Which reporter do you want to use? (Press <space> to select, <a> to toggle all
, <i> to invert selection)
>(*) spec
< > dot
< > junit
< > allure
< > sumologic
< > concise
< > reportportal
<Move up and down to reveal more choices>
```

Gambar 3.14 Pemilihan reporter

- f. Penulis menggunakan Chromedriver untuk menjalankan pengujian

```
Which reporter do you want to use? spec
? Do you want to add a service to your test setup? (Press <space> to select, <a>
to toggle all, <i> to invert selection)
>(*) chromedriver
< > sauce
< > testingbot
< > selenium-standalone
< > devtools
< > applitools
< > browserstack
<Move up and down to reveal more choices>
```

Gambar 3.15 Pemilihan Chromedriver

- g. Penulis menjalankan pengujian pada localhost:3000

```
? Do you want to add a service to your test setup? chromedriver
? What is the base url? http://localhost
Installing wdio packages:
- @wdio/local-runner
- @wdio/mocha-framework
- @wdio/spec-reporter
- wdio-chromedriver-service
- chromedriver
npm WARN registry Using stale data from https://registry.npmjs.org/ because the
host is inaccessible -- are you offline?
npm WARN registry Using stale data from https://registry.npmjs.org/ due to a req
uest error during revalidation.
[.....] / rollbackFailedOptional: WARN registry Using stale data f
```

Gambar 3.16 Penentuan URL Localhost

- h. Komponen yang akan terinstal adalah @wdio/local-runner, @wdio/mocha-framework, @wdio/spec-reporter, wdio-chromedriver-service dan chromedriver. Setelah selesai, maka WebDriverIO sudah dapat digunakan dengan menggunakan command npx wdio wdio.conf.js