

**PEMANFAATAN SELENIUM WEBDRIVER UNTUK  
PENGUJIAN REGRESI APLIKASI BERBASIS WEB (STUDI  
KASUS: *WEBSITE* PENYEDIA LAYANAN *ICON*)**



Disusun Oleh:

N a m a : Ratu Fairuz Hasna Sofani  
NIM : 19523079

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2023**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PEMANFAATAN SELENIUM WEBDRIVER UNTUK  
PENGUJIAN REGRESI APLIKASI BERBASIS WEB (STUDI  
KASUS: *WEBSITE* PENYEDIA LAYANAN *ICON*)**

**TUGAS AKHIR JALUR MAGANG**



N a m a  
NIM

: Ratu Fairuz Hasna Sofani  
: 19523079

الجامعة الإسلامية  
الاندونيسية  
Yogyakarta, 13 Juli 2023

Pembimbing,

( Moh. Idris, S.Kom., M.Kom. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**PEMANFAATAN SELENIUM WEBDRIVER UNTUK  
PENGUJIAN REGRESI APLIKASI BERBASIS WEB (STUDI  
KASUS: *WEBSITE* PENYEDIA LAYANAN *ICON*)**

**TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 13 Juli 2023

Tim Penguji

Moh. Idris, S.Kom., M.Kom.

**Anggota 1**

Lizda Iswari, S.T., M.Sc.

**Anggota 2**

Arrie Kurniawardhani, S.Si., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Ratu Fairuz Hasna Sofani  
NIM : 19523079

Tugas akhir dengan judul:

**PEMANFAATAN SELENIUM WEBDRIVER UNTUK  
PENGUJIAN REGRESI APLIKASI BERBASIS WEB (STUDI  
KASUS: *WEBSITE PENYEDIA LAYANAN ICON*)**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Juli 2023



( Ratu Fairuz Hasna Sofani )

## **HALAMAN PERSEMBAHAN**

Tugas akhir ini dengan tulus saya persembahkan untuk kedua orang tua saya yang sangat luar biasa. Untuk abah dan mama yang selalu mendukung baik secara moral maupun material, yang tidak pernah menekan saya untuk segera lulus, dan yang selalu mendo'akan saya dimanapun dan kapanpun. Tugas akhir ini juga saya persembahkan untuk diri saya sendiri yang telah berjuang hingga sampai pada titik ini.

## HALAMAN MOTO

لَا حَوْلَ وَلَا قُوَّةَ إِلَّا بِاللَّهِ الْعَلِيِّ الْعَظِيمِ

“Dan apabila hamba-hamba-Ku bertanya kepadamu tentang Aku, maka (jawablah), bahwasanya Aku adalah dekat. Aku mengabulkan permohonan orang yang berdoa apabila ia memohon kepada-Ku, maka hendaklah mereka itu memenuhi (segala perintah-Ku) dan hendaklah mereka beriman kepada-Ku, agar mereka selalu berada dalam kebenaran.”

– QS. Al-Baqarah : 186

*“Get a diploma.”* – Choi Seungcheol

*“Personally, I think that you shouldn't compare yourself with others. Just continue to work hard and someday the time will come that others will recognize your hard work.”*

– Jeon Wonwoo

*“If you arrive earlier than others, it may be nice, but going slower than others is not something wrong. I want to say that you're doing fine because you're walking on your own path.”* – Lee Seokmin

## KATA PENGANTAR

Alhamdulillah rabbil 'aalamiin, Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan petunjuk-Nya sehingga penulis dapat menyelesaikan laporan akhir dengan judul “Pemanfaatan Selenium WebDriver untuk Pengujian Regresi Aplikasi Berbasis Web (Studi Kasus: *Website* Penyedia Layanan *Icon*). Tidak lupa sholawat serta salam kepada junjungan kita Nabi Muhammad SAW yang telah menjadi teladan dalam setiap perjuangan dan usaha.

Laporan ini disusun sebagai syarat kelulusan penjaluran magang pada Program Studi Informatika - Program Sarjana. Adapun dalam penyusunan laporan akhir ini, penulis mendapatkan banyak dukungan, bimbingan, serta bantuan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua yang selalu memberikan dukungan penuh, nasehat, serta do'a tanpa batas.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika dan Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika - Program Sarjana.
3. Bapak Moh. Idris, S.Kom., M.Kom. selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk membimbing dan mengarahkan penulis sejak pelaksanaan kegiatan magang hingga selesai disusunnya laporan ini.
4. Seluruh pimpinan PT. Javan Solusi Indonesia yang telah memberikan kesempatan dan pengalaman magang.
5. Tim QA PT. Javan Solusi Indonesia terlebih Mba Anita, Mba Rahmi, Mas Yanuar, Mas Bram, serta *product owner website* Iconhub yaitu Mas Rizal yang telah membimbing, memberikan dukungan dan ilmu baru, serta membantu penulis akan banyak hal selama magang.
6. Para Dosen Program Studi Informatika - Program Sarjana yang telah memberikan bekal ilmu kepada penulis selama duduk di bangku perkuliahan.
7. Teman seperjuangan kuliah dan magang terutama Adis, Caca, Ilham, Okto, Sallu, Niken, dan Aksa yang saling membantu, memberi semangat, dan meluangkan waktu untuk mendengarkan keluh kesah selama perkuliahan.
8. Choi Seungcheol, Yoon Jeonghan, Hong Jisoo, Moon Junhui, Kwon Soonyoung, Jeon Wonwoo, Lee Jihoon, Xu Minghao, Kim Mingyu, Lee Seokmin, Boo Seungkwan,

Chwe Hansol, Lee Chan atas semua karya yang memberikan semangat serta menjadi *mood booster* bagi penulis selama penyusunan laporan akhir ini.

9. *Last but not least, I wanna thank me. I wanna thank me for believing in me, I wanna thank me for doing all this hard work, I wanna thank me for never quitting.*

Atas bantuan dari berbagai pihak, penulis dapat menyelesaikan laporan akhir ini. Akan tetapi laporan ini masih jauh dari kata sempurna, sehingga penulis menerima segala bentuk saran untuk menyempurnakan laporan ini. Akhir kata, semoga laporan ini dapat memberikan manfaat bagi para pembaca. Terima kasih.

Yogyakarta, 13 Juli 2023



( Ratu Fairuz Hasna Sofani )



## SARI

Pengujian perangkat lunak merupakan tahap penting dalam pengembangan perangkat lunak yang bertujuan untuk memverifikasi kinerja dan kesesuaian aplikasi atau sistem dengan kebutuhan pengguna. Perangkat lunak yang sedang dikembangkan tidak akan terhindar dari *bug* karena perangkat lunak akan terus dilakukan pembaharuan dan/atau perubahan pada sistem, sehingga dibutuhkan pengujian regresi untuk memastikan bahwa perubahan yang dilakukan selama proses pengembangan perangkat lunak tidak mempengaruhi fungsionalitas sistem yang sudah berjalan dengan baik sebelumnya.

Pengujian regresi merupakan sebuah pengujian yang menyeluruh, sehingga dalam pelaksanaannya membutuhkan banyak waktu jika dilakukan secara manual. Dalam beberapa tahun terakhir, pengujian otomatis telah menjadi metode yang populer dan efisien dalam melaksanakan pengujian perangkat lunak terutama pada pengujian regresi yang sifatnya menyeluruh dan berulang. Dengan menggunakan *automation tool* seperti Selenium WebDriver maka pengujian regresi dapat dilakukan secara menyeluruh dan tidak memakan banyak waktu.

Selenium WebDriver tidak memiliki fungsi untuk menghasilkan laporan hasil pengujian sendiri, sehingga dalam pengujiannya Selenium WebDriver membutuhkan implementasi dari alat pihak ketiga atau *third party tools* seperti Allure Framework Report. Dalam hal ini, Allure Framework Report membantu menghasilkan laporan hasil pengujian dengan sangat informatif.

Pada laporan akhir ini didapatkan hasil yaitu pengujian regresi dengan alat pengujian otomatis Selenium WebDriver dapat dilaksanakan secara cepat dan tepat serta penggunaan Allure Framework Report yang sangat informatif dapat mempermudah pembacaan hasil pengujian.

Kata kunci: Pengujian Perangkat Lunak, Pengujian Otomatis, Pengujian Regresi, Selenium WebDriver, Allure Framework Report

## GLOSARIUM

<i>Bug</i>	kesalahan atau kecacatan yang terjadi pada perangkat lunak.
<i>Icon</i>	tampilan berupa symbol yang menggambarkan fungsi objek itu sendiri.
<i>Scrum</i>	metode pengembangan perangkat lunak.
<i>Sprint</i>	aktivitas pekerjaan yang harus diselesaikan saat menggunakan metode <i>Scrum</i> .
<i>Test Case</i>	dokumen pengujian berisi skenario alur pengujian yang runtut.

## DAFTAR ISI

HALAMAN JUDUL .....	1
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	2
HALAMAN PENGESAHAN DOSEN PENGUJI.....	3
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	4
HALAMAN PERSEMBAHAN .....	5
HALAMAN MOTO .....	6
KATA PENGANTAR .....	7
SARI .....	9
GLOSARIUM.....	10
DAFTAR ISI.....	11
DAFTAR TABEL.....	1
DAFTAR GAMBAR .....	2
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Ruang Lingkup.....	2
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Sistematika Penulisan .....	4
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA .....	6
2.1 <i>Scrum</i> .....	6
2.2 Pengujian Perangkat Lunak .....	7
2.3 Pengujian Otomatis.....	7
2.4 Pengujian Regresi .....	8
2.5 Selenium .....	8
2.6 Tinjauan Pustaka.....	9
BAB III PELAKSANAAN MAGANG.....	12
3.1 <i>Onboarding</i> Magang.....	12
3.2 Manajemen Proyek .....	12
3.3 Sprint Ke-1.....	14
3.3.1 Analisis <i>Website</i> Iconhub.....	14
3.3.2 Pembuatan Skenario Pengujian.....	15
3.3.3 Pengujian Regresi Secara Manual .....	29
3.4 Sprint Ke-2 dan Ke-3 .....	30
3.4.1 Instalasi Selenium .....	31
3.4.2 Pembuatan Skrip Tes .....	31
3.4.3 Eksekusi Skrip Tes.....	34
3.4.4 Integrasi Allure Framework Report dengan Selenium WebDriver .....	67
3.4.5 Instalasi dan Eksekusi Skrip Tes dengan Allure Framework Report .....	68
3.5 Penutupan Proyek .....	71
BAB IV REFLEKSI PELAKSANAAN MAGANG.....	72
4.1 Relevansi Akademik .....	72
4.1.1 <i>Scrum</i> .....	72
4.1.2 Pengujian Regresi .....	72
4.1.3 Pengujian Otomatis menggunakan Selenium WebDriver .....	73
4.2 Pembelajaran Magang.....	73
BAB V PENUTUP .....	76
5.1 Kesimpulan .....	76
5.2 Saran .....	76
DAFTAR PUSTAKA .....	77

LAMPIRAN..... 79

## DAFTAR TABEL

Tabel 3.1 Pembagian dan deskripsi pekerjaan .....	13
Tabel 3.2 <i>Task</i> acuan pembuatan skenario pengujian.....	16
Tabel 3.3. Hasil pembuatan skenario pengujian .....	21
Tabel 3.4 Hasil pengujian regresi dengan Selenium WebDriver.....	59

## DAFTAR GAMBAR

Gambar 2.1 Alur kerja metode Scrum .....	6
Gambar 3.1 Dokumentasi <i>Sprint Review</i> .....	13
Gambar 3.2 Bukti penugasan pada Taiga untuk pembuatan dokumen <i>user guide</i> .....	14
Gambar 3.3 Bukti pengujian regresi secara manual pada QATouch.....	29
Gambar 3.4 Bukti penugasan pada Taiga .....	31
Gambar 3.5 Perintah instalasi Selenium .....	31
Gambar 3.6 Kode pengimporan fungsi, kelas dan objek .....	32
Gambar 3.7 Penulisan <code>'describe'</code> dan blok <code>'it'</code> .....	33
Gambar 3.8 Skrip tes TC0076 .....	34
Gambar 3.9 Skrip tes TC0077, TC0078, dan TC0080 .....	36
Gambar 3.10 Skrip tes TC0079 .....	38
Gambar 3.11 Skenario tes TC0081 .....	40
Gambar 3.12 Skrip tes TC0082 .....	41
Gambar 3.13 Skrip tes TC0083 .....	42
Gambar 3.14 Skrip tes TC0084 dan TC0085.....	43
Gambar 3.15 Skrip tes TC0086 .....	44
Gambar 3.16 Skrip tes TC0087 .....	45
Gambar 3.16 Skrip tes TC0088 .....	46
Gambar 3.17 Skenario tes TC0096 dan TC0097 .....	48
Gambar 3.18 Skrip tes TC0112 .....	51
Gambar 3.19 Skrip tes TC0116 .....	52
Gambar 3.20 Skrip tes TC0118 .....	53
Gambar 3.21 Skrip tes TC0119 .....	55
Gambar 3.22 Skrip tes TC0121 .....	56
Gambar 3.23 Perintah menjalankan salah satu skrip tes.....	56
Gambar 3.24 Tampilan halaman website Iconhub yang sedang diuji secara otomatis .....	57
Gambar 3.25 Hasil pengujian salah satu skrip tes .....	57
Gambar 3.26 Hasil pengujian salah satu skrip tes dengan banyak kasus uji .....	58
Gambar 3.27 Hasil pengujian satu folder skrip tes .....	58
Gambar 3.28 Perintah instalasi <i>dependencies allure-report</i> .....	68
Gambar 3.29 Penulisan skrip pada <i>file package.json</i> .....	68
Gambar 3.30 Tampilan terminal VSCode .....	68

Gambar 3.31 Tampilan awal halaman Allure Report pada <i>browser</i> .....	69
Gambar 3.32 Tampilan halaman Overview hasil pengujian regresi pada Allure Report .....	69
Gambar 3.33 Tampilan halaman Suites hasil pengujian regresi pada Allure Report .....	70

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

PT. Javan Cipta Solusi adalah perusahaan yang bergerak di bidang teknologi informasi dan memberikan layanan pengembangan perangkat lunak. Layanan yang disediakan oleh PT. Javan Cipta Solusi seperti kustom pengembangan perangkat lunak, integrasi sistem, layanan manajemen proyek, dan pengujian perangkat lunak. Perusahaan ini telah bekerja dengan berbagai klien dari berbagai industri, termasuk telekomunikasi, *e-commerce*, bahkan pemerintahan. Saat ini PT. Javan Cipta Solusi memiliki kantor yang berada di Perum Sukoharjo Indah J100 RT 11/RW 16, Kalurahan Sukoharjo, Kapanewon Ngaglik, Kabupaten Sleman, Yogyakarta serta memiliki dua tim *branchless* yang berada di Jakarta dan Bandung. Perusahaan memiliki visi yaitu “Menjadi mitra optimalisasi proses bisnis yang terdepan, berkelanjutan, dan menguntungkan.”

Pada kesempatan kali ini penulis ditempatkan untuk magang pada Divisi *Quality Assurance* (QA) yang beranggotakan 11 orang dengan seorang Lead QA yang bernama Anita Nur Hidayati. QA memiliki peran yang sangat penting dalam proses pengembangan perangkat lunak karena dapat membantu meningkatkan kualitas perangkat lunak, menjaga kepuasan pengguna, meningkatkan efisiensi, dan menjaga reputasi perusahaan. QA bertanggung jawab untuk memastikan bahwa perangkat lunak yang dibangun memenuhi standar kualitas yang ditentukan dan sesuai dengan persyaratan yang telah dibuat. Pada PT. Javan Cipta Solusi, seorang QA memiliki tugas wajib yaitu melakukan *unit testing* dan *regression testing*, melakukan pembuatan *automated testing* dengan Selenium WebDriver menggunakan bahasa JavaScript, kemudian ada beberapa tugas tambahan seperti mengikuti UAT atau *User Acceptance Testing*, mengikuti *daily meet* maupun *weekly sprint*, dan membuat dokumen *user guide* jika diminta.

Selama menjadi QA pada PT Javan Cipta Solusi, penulis telah ikut andil dalam beberapa proyek yang ada baik dari lembaga swasta maupun pemerintah. Beberapa proyek tersebut diantaranya adalah *compliance online system* (COS), sistem informasi manajemen pinjaman (SIMLOAN), sistem informasi manajemen kepegawaian (SIMPEG), dan sistem layanan penyedia *icon*.

Dalam penulisan laporan akhir ini penulis mengambil topik tentang pemanfaatan *automated testing tools* yang bernama Selenium WebDriver dengan bahasa Javascript pada



salah satu proyek yaitu pada pengembangan *website* penyedia layanan *icon* yang bernama Iconhub. Iconhub dikembangkan dengan tujuan agar dapat menyalurkan karya-karya anak bangsa dalam lingkup *design* untuk dipasarkan secara lokal maupun internasional serta dapat memudahkan para Editor dan Designer dalam pembuatan konten yang membutuhkan *icon-icon* tertentu. Sama dengan proses pengembangan perangkat lunak lainnya, Iconhub juga memerlukan proses pengujian sebagai salah satu tahapan pengembangan. Maka dari itu, dibuatkan *script* pengujian otomatis agar dapat dilakukan pengujian regresi dengan mudah sebagai salah satu cara pemeliharaan *website* Iconhub.

## 1.2 Ruang Lingkup

Kegiatan magang pada PT. Javan Cipta Solusi dilaksanakan selama tiga bulan terhitung sejak tanggal 20 Maret 2023 hingga 25 Juni 2023. Selama magang, penulis mendapatkan kesempatan untuk berpartisipasi dalam beberapa proyek dari lembaga swasta maupun pemerintahan. Beberapa proyek tersebut diantaranya adalah *compliance online system* (COS), sistem informasi manajemen pinjaman (SIMLOAN), sistem informasi manajemen kepegawaian (SIMPEG), dan *website* layanan penyedia *icon*.

*Compliance online system* (COS) adalah sebuah sistem kepatuhan perusahaan yang didasari oleh standarisasi ISO 37001:2016 tentang sistem manajemen anti-suap. ISO 37001:2016 merupakan sebuah standar yang diimplementasikan dalam bentuk kebijakan atau regulasi internal dalam organisasi dengan tujuan untuk mengontrol dan mencegah tindakan suap (Wijaya et al., 2020). ISO 37001:2016 digunakan untuk membantu organisasi memerangi suap dan mempromosikan budaya bisnis yang etis (Wijaya et al., 2020), sehingga dengan itu dikembangkan sebuah sistem kepatuhan *online* anti-suap untuk Perusahaan Y. Pada proyek COS ini penulis membantu mentor sejak tanggal 23 Maret 2023 hingga 3 April 2023 dalam pembuatan *test case* Dashboard Grafik Laporan Sertifikasi serta melakukan *unit testing* dan *regression testing* pada *sprint* 3.2 dan 3.3.

Sistem informasi manajemen pinjaman (SIMLOAN) adalah sebuah sistem yang dikembangkan untuk memfasilitasi pengelolaan pinjaman pada Perusahaan Z. Sebelum SIMLOAN dikembangkan, Perusahaan Z masih menggunakan cara manual yaitu dengan Microsoft Excel untuk mengelola pinjaman sehingga terdapat kesulitan dalam melakukan *monitoring* terhadap pembayaran yang akan dan sudah jatuh tempo, kesulitan dalam memprediksi kebutuhan pembayaran cicilan pinjaman, dan belum dapat mengakomodasi satu pinjaman dari beberapa pihak yang menyalurkan dana untuk pinjaman sehingga dibutuhkan

sebuah sistem yang efisien, cepat, dan mudah dikelola dan dapat memfasilitasi pengelolaan pinjaman Perusahaan Z yaitu SIMLOAN. Pada proyek SIMLOAN penulis membantu dalam pembuatan dokumen *user guide* untuk administrator.

Sistem informasi manajemen kepegawaian (SIMPEG) adalah sebuah sistem yang digunakan untuk mengumpulkan, memproses, menganalisis, dan menyajikan data serta informasi yang dibutuhkan untuk mendukung administrasi dan manajemen kepegawaian. SIMPEG dari sebuah Lembaga X sudah dikembangkan sejak tahun 2016 dan mengalami keusangan sehingga dibutuhkan peningkatan performa dari sisi teknologi agar semakin lincah dan menggunakan teknologi terkini sehingga tidak tertinggal oleh perkembangan zaman. Maka dari itu, dilakukan pengembangan ulang untuk SIMPEG milik Lembaga X dan dibuatkan *script* pengujian otomatis agar dapat dilakukan pengujian regresi yang efektif dan efisien sebagai salah satu cara pemeliharaan SIMPEG tersebut. Pada proyek ini, penulis berperan untuk membuat *script* pengujian otomatis dengan Selenium WebDriver Bahasa Javascript dibagian fitur Jenis Hukuman Disiplin, Jenis Pekerjaan, dan Jenis Pemberhentian.

Iconhub merupakan hasil kerja sama antara PT. Javan Cipta Solusi dengan One Week Wonder, Javan sebagai penyedia *platform* sedangkan OWW sebagai penyedia *icon*. Iconhub dikembangkan dengan tujuan dapat menyalurkan karya-karya anak bangsa dalam lingkup *design* untuk dipasarkan secara lokal maupun internasional serta dapat memudahkan para Editor dan Desainer dalam pembuatan konten yang membutuhkan *icon-icon* tertentu. Penulis berpartisipasi pada proyek ini dimulai pada tanggal 4 April 2023 hingga 12 Mei 2023. Pada proyek ini penulis bertugas untuk membuat *test case* seluruh *sprint*, membuat dokumen *user guide*, serta melakukan *unit testing* dan *regression testing*-nya. Penulis juga ditantang oleh Project Manager untuk dapat membuat *automation test* dengan Selenium WebDriver untuk memudahkan pengujian regresi yang akan dilakukan kemudian hari.

### 1.3 Tujuan

Laporan Tugas Akhir ini bertujuan untuk memberikan gambaran detail tentang pemanfaatan *automation tool* Selenium WebDriver dengan bahasa pemrograman Javascript sebagai alat bantu pengujian regresi pada sebuah *website* penyedia layanan *icon* bernama Iconhub serta pengintegrasian *third party tools* bernama Allure Framework Report sebagai kerangka kerja penghasil laporan pengujian.

## 1.4 Manfaat

Manfaat yang diperoleh dari penulisan laporan tugas akhir ini adalah dapat mengetahui cara kerja Selenium WebDriver dengan Bahasa Javascript sebagai *automation tool* untuk pengujian regresi sebuah *website* penyedia layanan *icon* bernama Iconhub serta dapat membuktikan bahwa dengan menggunakan Selenium WebDriver maka pengujian regresi tidak membutuhkan waktu yang lama.

Sedangkan manfaat dari pengintegrasian *third party tool* bernama Allure Framework Report adalah sebagai pelengkap Selenium WebDriver dalam pekerjaannya sehingga dapat menghasilkan akurasi dan kinerja yang lebih baik. Pengintegrasian ini juga dapat menghasilkan laporan hasil pengujian yang mudah dimengerti dan mudah dibagikan, sehingga memudahkan kolaborasi dan pemantauan pengujian.

Hasil pengujian bermanfaat untuk memvalidasi fungsionalitas dari sebuah sistem perangkat lunak serta dapat mengidentifikasi kesalahan dan cacat yang perlu diperbaiki sebelum perangkat lunak dirilis.

## 1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan laporan ini adalah sebagai berikut:

a. BAB I: Pendahuluan

Bab ini berisi latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan Laporan Akhir.

b. BAB II: Landasan Teori dan Tinjauan Pustaka

Bab ini berisi landasan teori dan kajian pustaka dari penelitian-penelitian terdahulu yang berkaitan dengan topik Laporan Akhir.

c. BAB III: Pelaksanaan Magang

Bab ini berisi gambaran selama pelaksanaan magang dan tahapan serta *script* pengujian otomatis menggunakan Selenium WebDriver pada *website* layanan penyedia *icon* bernama Iconhub.

d. BAB IV: Refleksi Pelaksanaan Magang

Bab ini berisi relevansi akademik yang membahas terkait gap antara teori akademik dengan pelaksanaan di lapangan selama magang serta pembelajaran magang yang didapatkan dari PT. Javan Cipta Solusi.

e. BAB V: Kesimpulan

Bab ini berisi kesimpulan dari hasil yang didapatkan setelah melakukan pengujian otomatis dengan Selenium WebDriver Bahasa Javascript sebagai alat bantu pengujian regresi untuk sistem informasi manajemen kepegawaian (SIMPEG).

## BAB II

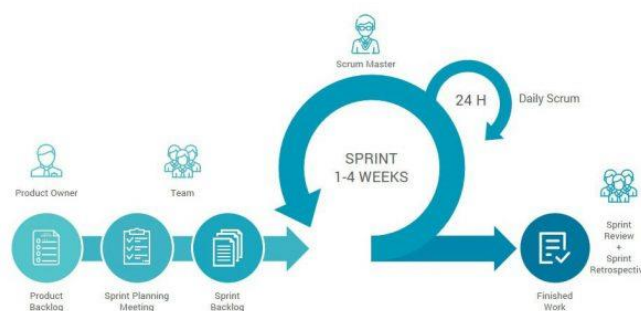
### LANDASAN TEORI DAN TINJAUAN PUSTAKA

#### 2.1 Scrum

Salah satu metode dalam pengembangan perangkat lunak adalah *Scrum*. *Scrum* adalah metode pengembangan dan manajemen yang mengikuti prinsip-prinsip metodologi *Agile* dikarenakan *Scrum* sendiri adalah metode turunan dari *Agile*. *Scrum* merupakan metode yang digunakan untuk mengembangkan perangkat lunak serta mengelola produk atau aplikasi yang sedang dikembangkan (Permana, 2015). Pendekatan *Scrum* berfokus pada strategi dan pengembangan produk yang fleksibel dimana tim pengembang bekerja sebagai satu kesatuan untuk mencapai tujuan bersama.

Pada buku berjudul “*Process Improvement and CMMI for Systems and Software: Planning, Implementation, and Management*” yang ditulis oleh Kenett & Baker (2010), dikatakan bahwa dalam proses pengembangan menggunakan metode *Scrum* terdapat beberapa *key practices* (kunci praktek), yaitu berpegang pada jadwal, bekerja konsisten sesuai *sprint*, *product backlog* merupakan tanda dalam memulai pekerjaan, *product backlog* merupakan dasar dari melakukan *sprint* dan tim harus memutuskan produk dapat dikembangkan atau tidak, *scrum master* bertanggung jawab menerima hasil *sprint*, mengadakan pertemuan setiap hari kerja, serta berfokus pada *sprint*, pertemuan, *review*, dan jadwal proyek.

Kemudian pada buku berjudul “*A Practical Guide to Distributed Scrum*” karya Woodward et al., (2010), disebutkan ada tiga peran dalam *Scrum*, yaitu *product owner*, *scrum master*, dan tim. *Product owner* merupakan seseorang yang menentukan spesifikasi dan bisnis dari aplikasi yang akan dibangun. Tim adalah orang-orang yang menjalankan proyek, tim dapat terdiri dari beberapa peranan lain seperti sistem analis, bisnis analis, penguji, dan programmer. Sedangkan *scrum master* adalah seseorang yang mengatur proses *Scrum* selama proyek berlangsung.



Gambar 2.1 Alur kerja metode Scrum

## 2.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses verifikasi dan validasi perangkat lunak dengan memeriksa apakah memenuhi persyaratan pengguna dan apakah semuanya diterapkan sesuai dengan standarnya (Anand & Uddin, 2019). Pengujian perangkat lunak memainkan peran penting dalam proses pengembangan perangkat lunak berkualitas tinggi. Pengujian perangkat lunak harus dilakukan dengan tepat dan serius agar menghindari kerugian finansial dan reputasi yang dapat timbul akibat perangkat lunak tidak berjalan dengan baik.

Mengacu pada jurnal berjudul “Importance of Software Testing in the Process of Software Development” karya Azeem Uddin dan Abhineet Anand, disebutkan empat level atau tingkat pengujian yang dikenal dan digunakan secara luas oleh para praktisi pengujian perangkat lunak yaitu *Unit Testing*, *Integration Testing*, *System Testing*, dan *Acceptance Testing*. *Unit Testing* adalah jenis pengujian perangkat lunak yang dilakukan pada unit-unit terkecil dari sebuah perangkat lunak. *Unit Testing* dilakukan dengan tujuan untuk menemukan kesalahan atau *bug* dalam kode dan memastikan bahwa unit-unit tersebut berfungsi dengan baik sebelum digabungkan dengan bagian lain perangkat lunak. *Integration Testing* adalah pengujian yang melibatkan pengujian interaksi antara unit-unit yang telah diuji pada level *Unit Testing*. *Integration Testing* bertujuan untuk memastikan bahwa unit-unit tersebut dapat berinteraksi satu sama lain dan membentuk sistem yang utuh. *System Testing* adalah pengujian keseluruhan sistem, pengujian ini dilakukan dengan skenario pengujian yang mencakup fitur dan fungsionalitas sistem dan meliputi pengujian fungsional dan non-fungsional. Yang terakhir, *Acceptance Testing* adalah pengujian yang dilakukan oleh pengguna dengan tujuan apakah perangkat lunak sudah sesuai dengan persyaratan bisnis dan apakah sudah siap digunakan oleh pengguna.

## 2.3 Pengujian Otomatis

Pengujian perangkat lunak dapat dilakukan dengan dua cara, yaitu secara manual dan otomatis. Pengujian manual adalah pengujian yang dilakukan langsung oleh manusia tanpa alat bantuan. Pengujian otomatis adalah pengujian yang dilakukan oleh manusia dengan bantuan alat penguji otomatis untuk menjalankan skenario pengujian. Pengujian dilakukan dengan cara menulis skenario pengujian manual dalam bentuk skrip atau kode yang dapat dijalankan oleh alat bantu otomatis (Hanna et al., 2018). Pengujian otomatis dapat menghemat waktu dan usaha serta meningkatkan efektivitas pengujian (Hanna et al., 2018). Selain itu, pengujian otomatis

juga memungkinkan pengujian dilakukan di lingkungan yang sama, sehingga hasil pengujian menjadi lebih konsisten dan dapat diulang. Dalam pengujian otomatis, pengujian dapat dilakukan dengan alat bantu pengujian seperti Selenium, Katalon, Playwright, Cypress, dan lain sebagainya.

## 2.4 Pengujian Regresi

Pengujian regresi adalah jenis pengujian perangkat lunak yang memiliki tujuan untuk memastikan bahwa perubahan yang dibuat pada saat pengembangan perangkat lunak tidak mempengaruhi fitur perangkat lunak yang seharusnya tidak berubah (Wong et al., 1997). Dalam pengujian regresi, pengembang perangkat lunak melakukan pengujian pada bagian-bagian sistem yang terdampak oleh perubahan serta bagian-bagian yang tidak terdampak. Teknik tersebut dilakukan untuk menguji modifikasi pada versi perangkat lunak sebelumnya, sehingga dapat mengetahui dimana letak perubahan dan bagian-bagian yang seharusnya tidak berubah.

Jurnal berjudul *A Study of Effective Regression Testing in Practice* karya Eric Wong dkk membuah hasil bahwa pengujian regresi yang efektif memerlukan perencanaan yang matang, penggunaan teknik pengurangan *test case*, pengujian prioritas yang disesuaikan dengan perubahan terbaru pada perangkat lunak, serta bahwa manajemen konfigurasi dan alat bantu otomatis termasuk dalam faktor kunci pengujian regresi yang efektif (Wong et al., 1997).

## 2.5 Selenium

Selenium merupakan sebuah perangkat lunak *open-source* yang digunakan untuk melakukan pengujian otomatis perangkat lunak pada berbagai platform dan sistem operasi (Vila et al., 2017). Selenium biasa digunakan untuk pengujian otomatis pada aplikasi web, tetapi dapat pula digunakan untuk pengujian pada aplikasi *desktop* maupun *mobile*. Selenium mendukung berbagai macam bahasa pemrograman seperti Java, Python, C#, dan lain sebagainya sehingga memudahkan pengujian perangkat lunak pada lingkungan pengembangan. Selenium terdiri dari beberapa komponen, yaitu Selenium IDE, Selenium WebDriver, Selenium Grid, dan Selenium Standalone Server yang dapat digunakan secara terpisah maupun digabungkan untuk memenuhi kebutuhan pengujian perangkat lunak.

Selenium IDE adalah perangkat lunak untuk pengujian otomatis pada aplikasi web yang bekerja sebagai ekstensi pada *browser* seperti Mozilla Firefox dan Google Chrome. Selenium IDE bekerja dengan cara merekam dan memutar kembali interaksi pengguna dengan aplikasi

web secara otomatis atau biasa disebut dengan metode pengujian *Record and Playback Testing*. Selenium IDE menyediakan fitur untuk merekam, mengedit, dan melakukan *debugging* selama proses pengujian, sehingga penguji dapat fokus pada pengembangan kasus uji (Prabowo D & Putro H, 2019). Selenium IDE mudah digunakan karena memiliki tampilan GUI (Graphical User Interface) yang sederhana dan memiliki perintah yang sedikit sehingga mudah dipelajari.

Selenium WebDriver merupakan API (*Application Programming Interface*) yang memungkinkan pengguna untuk melakukan pengujian otomatis perangkat lunak melalui pengkodean dengan bahasa pemrograman seperti Java, Python, C#, dan lain sebagainya. Selenium WebDriver memungkinkan pengguna untuk berkomunikasi langsung dengan *browser*. Selenium WebDriver mendukung berbagai *browser* web seperti Google Chrome, Mozilla Firefox, Microsoft Edge, dan lain sebagainya (Ramya P et al., 2017).

Selenium Grid adalah alat pengujian otomatis untuk aplikasi web yang memungkinkan pengguna untuk menjalankan tes di waktu yang bersamaan pada beberapa mesin dan browser yang berbeda. Selenium Grid dapat digunakan untuk memecah pengujian yang panjang ke dalam beberapa bagian yang lebih kecil. Bagian-bagian yang lebih kecil itu kemudian dijalankan secara bersamaan pada beberapa mesin dan *browser* yang berbeda sehingga dapat meningkatkan kecepatan pengujian (Bruns A et al., 2009)

Selenium Standalone Server merupakan komponen dasar dari Selenium yang digunakan untuk menghubungkan Selenium WebDriver dan Selenium Grid ke browser web yang akan diuji. Selenium Standalone Server mendukung berbagai bahasa pemrograman dan *platform* seperti Windows, Mac, dan Linux.

## 2.6 Tinjauan Pustaka

Terdapat beberapa penelitian sebelumnya yang telah mengkaji terkait Selenium WebDriver. Dalam penelitian García et al., (2020), dilakukan survei terhadap ekosistem Selenium yang mencakup Selenium WebDriver, Selenium IDE, Selenium Grid, dan berbagai pustaka Selenium yang dapat digunakan untuk bahasa pemrograman tertentu. Survei yang dilakukan oleh 72 orang dari 24 negara pada tahun 2019 tersebut menghasilkan kesimpulan bahwa sebanyak 98,61% dari mereka menggunakan Selenium WebDriver dan 50% memilih menggunakan bahasa pemrograman Java, yang kemudian disusul Python, dan JavaScript. Pada Java, *framework* yang biasa digunakan adalah JUnit sebanyak 75% dan TestNG sebanyak 52,78%. Pada Python, *framework* yang biasa digunakan adalah PyTest dan pada JavaScript



separuh penguji memilih menggunakan *framework* Mocha. Untuk alat pelaporan hasil uji, sebanyak 62,5% responden tidak menggunakannya dan 11,11% memilih Allure.

Kemudian pada penelitian yang melakukan analisis serta perancangan *framework* pengujian otomatis berbasis Selenium WebDriver (Gojare et al., 2015), disebutkan bahwa Selenium WebDriver tidak memiliki fungsi untuk menghasilkan laporan atau *report* untuk kasus uji yang gagal. Selenium WebDriver tidak memiliki kemampuan bawaan untuk menghasilkan hasil uji. Maka dari itu untuk mengetahui laporan hasil uji Selenium WebDriver memerlukan alat pihak ketiga (*third party tools*). Dengan menggunakan *framework* ini, penguji dapat menghasilkan laporan pengujian yang disesuaikan dan juga dapat menganalisis kegagalan menggunakan *screenshot* dari kasus pengujian yang gagal.

Dalam penelitian berjudul, “Automation Testing Framework for Web Applications with Selenium WebDriver: Opportunities and Threats” (Vila et al., 2017), disebutkan beberapa keuntungan dan ancaman dari Selenium WebDriver. Keuntungan yang disebutkan dalam penelitian tersebut antara lain, Selenium adalah alat *open-source* sehingga tidak berbayar dan sangat fleksibel, mendukung banyak web *browser*, platform, dan bahasa pemrograman, Selenium WebDriver mendukung IDE dan kerangka pengujian, mensimulasikan pengguna bekerja dengan *browser* secara nyata, dapat melakukan pengujian pada web dengan halaman yang dinamis, dimungkinkan untuk membuat rangkaian pengujian otomatis yang lengkap, serta pemeliharaan yang tinggi dari skrip pengujian otomatis yang dibuat. Sedangkan ancaman yang disebutkan dalam penelitian tersebut antara lain, diperlukan kompetensi teknis yang tinggi dalam bahasa pemrograman, tidak dapat membuat kasus uji otomatis secara langsung, perlu terhubung dan bekerja dengan *framework* dan *add-on* pihak ketiga seperti TestNG, Firebug, FirePath, dan lain sebagainya, Selenium WebDriver cukup memakan waktu dan sumber daya, serta dapat menghasilkan lalu lintas jaringan yang lebih besar. Penelitian tersebut menghasilkan kesimpulan bahwa keuntungan dari Selenium WebDriver lebih unggul daripada ancaman yang ada. Selenium tetap menjadi alat yang paling banyak digunakan dan disukai untuk membuat *framework* pengujian otomatis. Selenium WebDriver menawarkan banyak kemungkinan untuk membuat pengujian fungsional. Selenium WebDriver fleksibel, dapat diperluas, dan dapat mensimulasikan perilaku pengguna yang sebenarnya.

Terdapat penelitian yang menguji sebuah *website* Lawyer’s Hub menggunakan Selenium WebDriver (Ramya P et al., 2017). Langkah-langkah yang dilakukan adalah menganalisis masalah yang ada dan penulisan *test cases*, setelah itu implementasi menggunakan Maven yang terintegrasi dengan Selenium WebDriver, kemudian dilakukan eksekusi dari *test case* yang

sudah dibuat kodenya. Hasil dari penelitian tersebut menyatakan bahwa salah satu keuntungan dari pengujian otomatis adalah pengujian perangkat lunak dapat digunakan kembali dan semua skrip pengujian dapat didokumentasikan dan dipelihara dengan mudah.

Berdasarkan penelitian yang telah disebutkan sebelumnya, penggunaan Selenium WebDriver untuk pengujian otomatis pada aplikasi web sangat efisien, sederhana, dan dapat menguji dengan akurasi yang baik. Tetapi jika didukung oleh alat bantu lain seperti *third party tools* atau *add-on* dalam integrasi dengan Selenium WebDriver itu sendiri maka akan menghasilkan kegunaan dan akurasi yang jauh lebih baik (Ramya P et al., 2017)

Adapun beberapa penelitian yang telah dikaji sebelumnya membahas terkait pengujian otomatis pada aplikasi berbasis web. Pada penelitian yang dilakukan oleh Amalia & Cahyono (2022) dikatakan bahwa yaitu dengan menguji sebuah sistem manajemen jaringan menggunakan Playwright didapatkan kesimpulan bahwa Playwright sangat mudah digunakan oleh penguji yang baru mengenal *automation testing* karena kemudahannya dalam melakukan pengujian, sedangkan Selenium WebDriver akan lebih mudah digunakan oleh penguji yang memiliki kemampuan *programming* yang handal. Sedangkan pada penelitian yang berjudul “*Automation Testing Tool dalam Pengujian Aplikasi The Point of Sale (Studi Kasus TPOS PT. JAVA SIGNA INTERMEDIA)*” (Kosasih & Cahyono, 2021) memiliki kesimpulan bahwa sebaiknya dibuatkan *automation script* untuk pengujian yang bersifat repetitif atau sistem yang memiliki data besar, tetapi jika hanya perlu menguji satu kali atau dibutuhkan pengalaman langsung sebaiknya dilakukan dengan manual.

Berdasarkan informasi yang disajikan dalam penelitian-penelitian tersebut maka akan digunakan untuk mendukung pengujian regresi perangkat lunak dengan Selenium WebDriver yang akan penulis lakukan. Penulis juga belum menemukan penelitian yang membahas tentang pengujian regresi menggunakan Selenium WebDriver sehingga memperkuat alasan penulis dalam menyusun laporan akhir mengenai pengujian regresi dengan Selenium WebDriver ini.

## BAB III

### PELAKSANAAN MAGANG

#### 3.1 *Onboarding* Magang

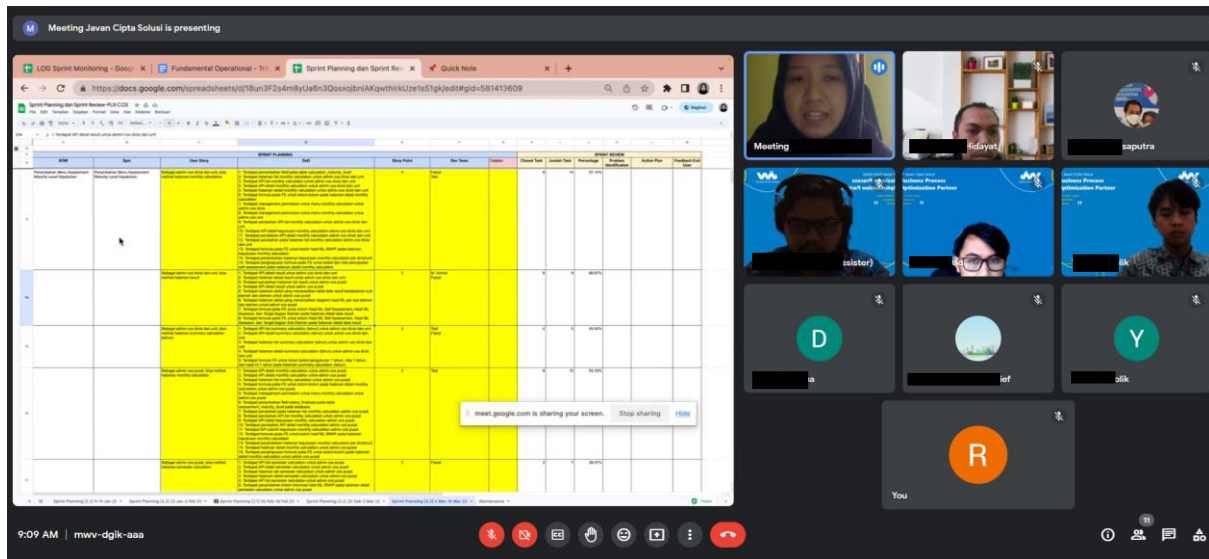
Tahapan pertama bagi para pemegang sebelum masuk ke dalam proyek-proyek yang ada adalah melaksanakan *onboarding* magang. Pada awal magang, penulis wajib menyelesaikan *task onboarding* terlebih dahulu. *Task onboarding* yang diberikan untuk penulis mencakup pembuatan resume tentang pemahaman mengenai perusahaan yang telah dijelaskan oleh HRD dan resume tentang pemahaman mengenai QA yang telah dijelaskan oleh Mentor. Pemahaman mengenai QA mencakup bagaimana cara menggunakan alat manajemen tes yang disebut QATouch dengan benar, seperti bagaimana membuat *test case*, bagaimana mengeksekusi *test run*, dan bagaimana membuat *issue* pada QATouch jika menemukan *bug* ketika mengeksekusi kasus uji. Kemudian pemahaman mengenai QA juga mencakup bagaimana cara membuat dokumen *user guide* yang baik, pemahaman tentang *matrix RACI* untuk QA pada perusahaan, dan bagaimana *reporting progress* oleh QA kepada Tim Project.

Setelah menyelesaikan *task onboarding* magang, pada waktu itu Tim QA dihadapkan dengan tantangan baru yaitu untuk mulai menerapkan pengujian otomatis pada beberapa proyek. Sebelum pengujian otomatis diterapkan untuk seluruh Tim QA, Lead QA menyelenggarakan sebuah uji kompetensi Selenium WebDriver. Uji kompetensi ini dilakukan untuk mengukur seberapa paham dan mampu Tim QA dalam membuat pengujian otomatis menggunakan Selenium WebDriver. Uji kompetensi ini dilaksanakan menggunakan Google Meet selama 30 menit dengan satu *task* yaitu membuat pengujian otomatis untuk melakukan *booking* perjalanan melalui salah satu *platform* travel. Kemudian setelah 30 menit berlalu, maka skrip kode yang telah ditulis harus dimasukkan ke dalam Gitlab khusus milik Tim QA.

#### 3.2 Manajemen Proyek

PT. Javan Cipta Solusi mengimplementasikan kerangka kerja *Scrum* dalam pengembangan proyeknya. Pada PT. Javan Cipta Solusi, *sprint* harus terintegrasi dengan 6 *weeks cycle*, yang berarti dalam satu siklus enam mingguan dibagi menjadi tiga *sprint* dan satu *sprint* dijalankan selama dua minggu. *Sprint* dimulai dengan *Sprint Planning* yang memiliki tujuan untuk menganalisis pekerjaan-pekerjaan yang akan masuk ke dalam *sprint* tersebut. Pada akhir *sprint*, dilakukan *Sprint Review* untuk mengevaluasi pekerjaan-pekerjaan yang telah

selesai. Sedangkan setiap harinya dilakukan *Daily Sprint* selama kurang lebih 15 menit untuk membahas perkembangan progres. Adapun dokumentasi pada saat penulis mengikuti *Sprint Review* ditunjukkan pada Gambar 3.1.



Gambar 3.1 Dokumentasi *Sprint Review*

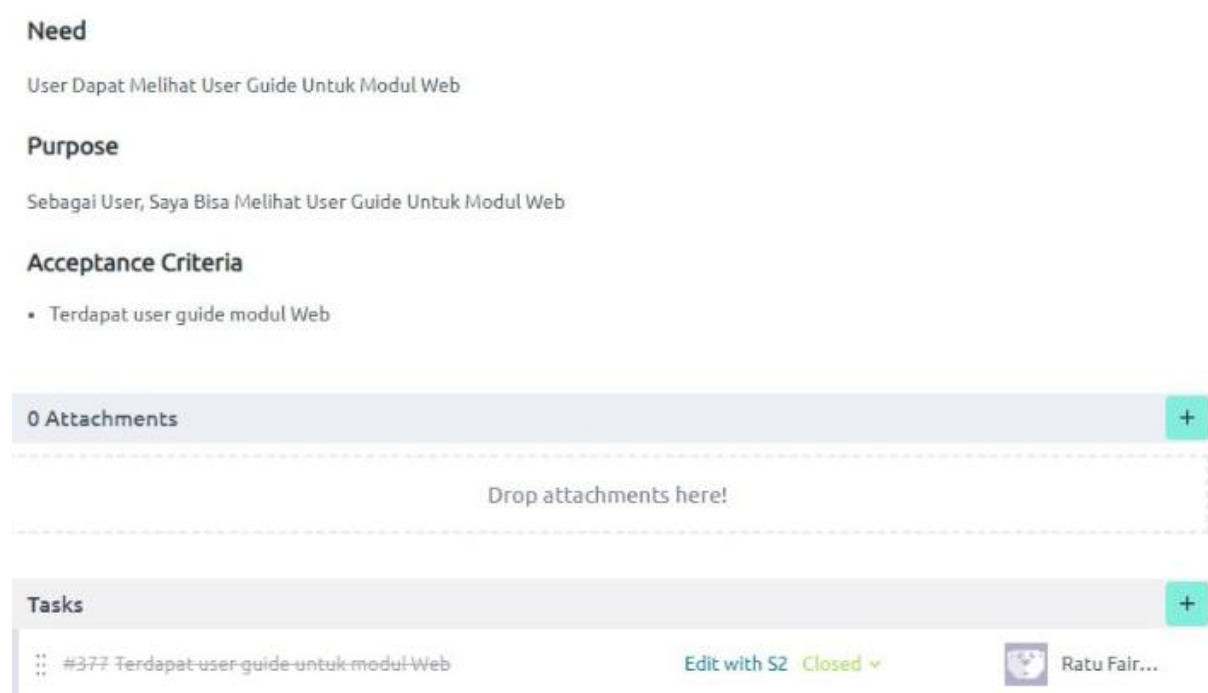
Alat manajemen proyek yang digunakan oleh PT. Javan Cipta Solusi bernama Taiga. Taiga merupakan sebuah perangkat lunak manajemen proyek untuk membantu mengorganisir dan mengelola pekerjaan-pekerjaan dalam proses pengembangan sistem. Semua *task* yang dikerjakan harus dituliskan ke dalam Taiga agar memudahkan dalam pengecekan progress. Setiap *task* diberikan status progres yang berfungsi sebagai pelacakan status pekerjaan. Pada Tabel 3.1 dijelaskan mengenai pembagian peran dalam proyek *website* Iconhub.

Tabel 3.1 Pembagian dan deskripsi pekerjaan

Nama	Role Pekerjaan	Deskripsi
Purwa D	Tribe Lead	Bertanggung jawab penuh atas proyek dan tim, menjadi jembatan komunikasi antar tim internal dan eksternal.
Muhammad Rizal Muttaqin	Project Owner dan System Analyst	Bertanggung jawab atas proyek, memastikan <i>sprint</i> berjalan dengan baik, menganalisis kebutuhan sistem.
Moh. Akbar	Programmer <i>Website</i> Iconhub	Melakukan pengembangan dan perbaikan sistem web.
Shaddam	Programmer <i>Website</i> Iconhub	Melakukan pengembangan dan perbaikan sistem web.
Fajri Idza	Programmer <i>Plugin</i> Iconhub	Melakukan pengembangan dan perbaikan pada <i>plugin</i> .
Rahmi Rahayu Fatihan	Quality Assurance	Melakukan pengujian sistem.
Ratu Fairuz Hasna Sofani	Quality Assurance	Melakukan pengujian sistem.

### 3.3 Sprint Ke-1

Penulis masuk ke proyek *website* Iconhub pada *sprint* ke-1 diawal bulan April, tepatnya pada tanggal 4 April 2023. Sprint ke-1 ini berlangsung pada tanggal 3 April 2023 hingga 14 April 2023. Pada saat itu penulis berkontribusi sebagai QA yang memiliki tanggung jawab untuk melakukan pengujian sistem dan memastikan bahwa fungsi pada sistem telah berjalan dengan baik. Pada sprint ke-1 ini penulis melakukan analisis *website* Iconhub, membuat *test case* atau skenario pengujian, melakukan pengujian regresi secara manual, serta membuat dokumen *user guide*. Pada Gambar 3.2 terdapat bukti penugasan pada Taiga untuk pembuatan dokumen *user guide*.



Gambar 3.2 Bukti penugasan pada Taiga untuk pembuatan dokumen *user guide*

#### 3.3.1 Analisis Website Iconhub

*Website* Iconhub merupakan sebuah aplikasi berbasis web yang menyediakan layanan *icon*. Pada Iconhub tersedia lebih dari 1000 macam *icon* yang dapat digunakan secara gratis untuk membantu kebutuhan desain. Ada tiga *role* pengguna yang dapat mengakses *website* ini, yaitu *guest user*, *member user*, dan admin. *Guest user* adalah pengguna bebas yang tidak melakukan *login* dengan akun mereka. Sebagai *guest user*, pengguna dapat mengakses halaman Iconhub sama seperti *member user* hanya saja ketika akan mengunduh *icon* akan ada sebuah modal konfirmasi untuk memasukkan email. Sedangkan jika mengakses halaman Iconhub

sebagai *member user*, yang mana harus melakukan *login* terlebih dahulu, ketika akan mengunduh *icon* tidak akan diminta untuk memasukkan email kembali.

*Role admin* adalah *role* pengguna yang paling berbeda dengan dua *role* lainnya yang sudah saya sebutkan sebelumnya. *Role admin* memiliki peranan khusus yaitu dapat mengakses menu *Dashboard* milik Iconhub, dan dapat melakukan *edit password* serta *edit profile*. Menu *Dashboard* berisikan beberapa menu-menu lainnya seperti *Manage Icon*, *Manage Category*, *User*, dan *Role*. Menu *Manage Icon* adalah menu dimana admin dapat mengelola *icon-icon* dalam Iconhub seperti menambah, mengedit, dan menghapus *icon*. Menu *Manage Category* juga hampir sama dengan menu *Manage Icon*, yaitu menu dimana admin dapat mengelola jenis-jenis kategori *icon* dalam Iconhub seperti menambah, mengedit, dan menghapus kategori.

Menu *User* yaitu menu untuk admin mengelola pengguna yang terdaftar pada *website* Iconhub. Pada menu ini, admin mendapatkan akses untuk melihat data pengguna seperti email yang digunakan untuk mendaftar, *role* dan status dari pengguna tersebut, dan kapan pengguna tersebut teregistrasi yang tertera pada halaman *list user*. admin dapat menambah pengguna baru secara manual melalui *button Add* yang ada pada menu *User* tersebut. Menu *Role* yaitu menu untuk admin mengelola *role* yang ada pada *website* Iconhub. Pada menu ini, admin dapat melihat jumlah pengguna berdasarkan *role*-nya, dapat menambah *role* baru, mengedit, serta menghapus *role* yang ada.

### 3.3.2 Pembuatan Skenario Pengujian

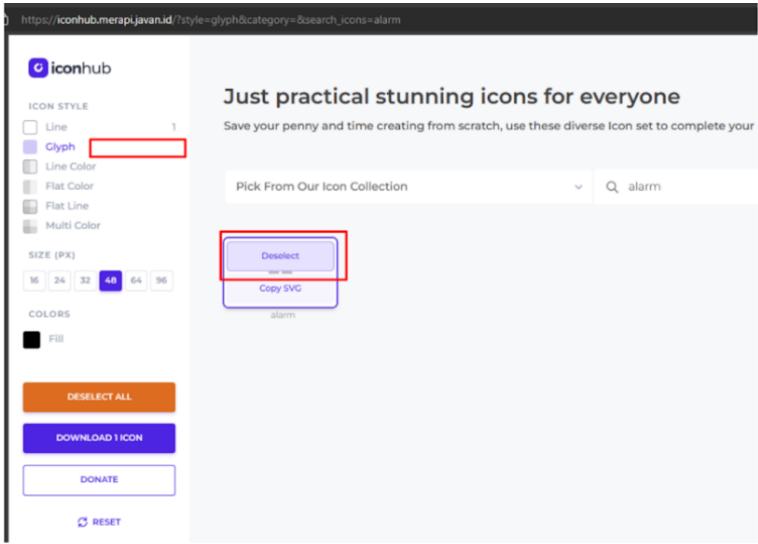
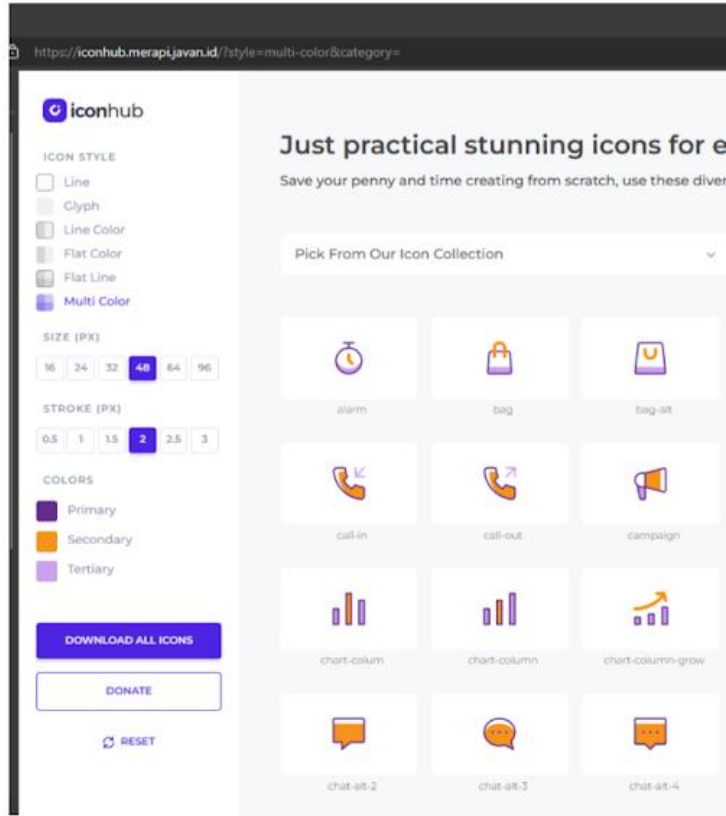
Skenario pengujian sangat penting dalam proses pengujian perangkat lunak. Skenario pengujian digunakan sebagai dokumen acuan dalam melakukan pengujian. Dokumen ini berisi langkah-langkah pengujian, hasil ekspektasi, hasil aktual, dan status pengujian. Terdapat dua jenis pengujian pada skenario uji, yaitu skenario uji positif dan skenario uji negatif. Skenario uji positif dibuat untuk memastikan sistem melakukan proses yang seharusnya dilakukan dan skenario uji negatif dibuat untuk memastikan sistem tidak melakukan proses yang tidak seharusnya dilakukan. Adapun penulis membuat skenario pengujian menggunakan alat manajemen tes bernama QATouch dikarenakan QATouch menyediakan fitur untuk mempermudah pendokumentasian skenario pengujian ini.

Pembuatan skenario pengujian dilakukan ketika sistem telah dianalisis oleh sistem analis dan *task*-nya sudah dibuat untuk kemudian dikerjakan oleh programmer. Pada *task* tersebut berisi deskripsi mengenai fungsi suatu fitur dan *acceptance criteria* dari fitur tersebut. Melalui *acceptance criteria* yang sudah dituliskan itu maka penulis dapat membuat skenario pengujiannya. Pada sprint ke-1 ini penulis melakukan pembuatan skenario pengujian sebanyak

total 72 buah yang mencakup keseluruhan modul pada *website* Iconhub, yaitu modul pada *role guest user, member user, dan admin*. Akan tetapi pada penulisan laporan akhir ini penulis berfokus pada modul *role member user* sebanyak 20 skenario pengujian. Pada Tabel 3.2 memperlihatkan acuan (*acceptance criteria/definition of done*) yang terdapat pada *task* yang dibuat oleh sistem analis bagi penulis untuk membuat skenario pengujian.

Tabel 3.2 *Task* acuan pembuatan skenario pengujian

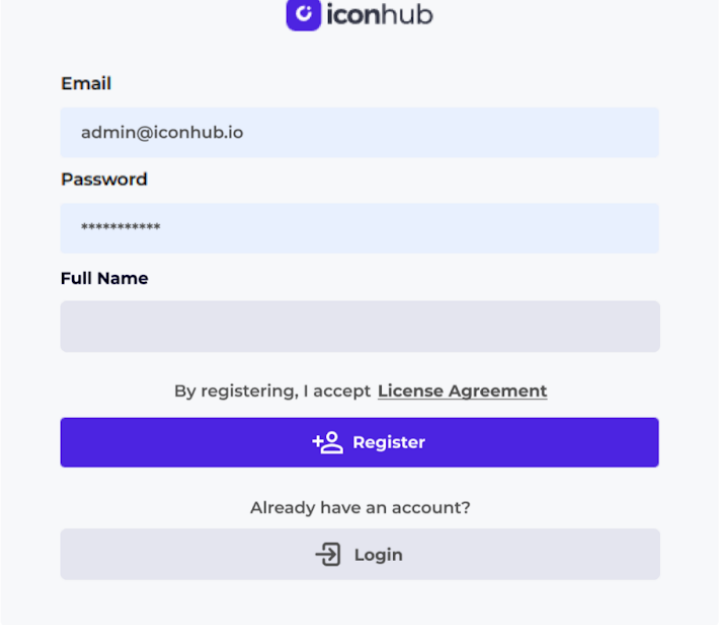
Kode Test Case	Skenario	Task Sistem Analis
TC0075	Dapat melakukan login sebagai member user	<p><b>Acceptance Criteria / Definition of Done?</b></p> <p>user melakukan login dengan menggunakan email dan password ketika berhasil maka konfirmasi email dan ketika gagal munculkan alasan gagal register nya</p>
TC0076	Dapat melakukan reset password sebagai member user	<p><b>Acceptance Criteria</b></p> <ul style="list-style-type: none"> <li>• Dapat mengirimkan email ubah password</li> <li>• User dapat klik link reset password dari email yang dikirimkan</li> <li>• User dapat menuju halaman reset password</li> <li>• Password user berhasil berubah di dalam database</li> </ul>
TC0077	Dapat memilih satu atau banyak icon per icon style sebagai member user	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Icon yang terpilih di icon style a dapat di 'select' kembali di icon style lainnya</li> <li>• Ketika didownload, nama dan format icon sesuai dengan yang diatur pada aplikasi</li> </ul>
TC0078	Dapat menunduh satu atau beberapa icon sebagai member user	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Icon yang terpilih dapat didownload dengan format svg dan zip memiliki nama file dengan format [style_name]_[icon_name]</li> </ul>

Kode Test Case	Skenario	Task Sistem Analisis
TC0079	Dapat <i>deselect</i> satu atau banyak <i>icon</i> per <i>icon style</i> sebagai <i>member user</i>	 <p>Expected Result</p> <ul style="list-style-type: none"> <li>• Icon yang terpilih di style line bisa di select kembali di syle glyph</li> </ul> <p>Definition of Done</p> <ul style="list-style-type: none"> <li>• Icon yang terpilih di style line bisa di select kembali di syle glyph</li> </ul>
TC0080	Dapat <i>deselect</i> seluruh <i>icon</i> yang dipilih sebagai <i>member user</i>	 <p>Definition of Done</p> <ul style="list-style-type: none"> <li>• Icon yang belum di pilih lagi, tulisannya harusnya "Select" dan tidak ada kotak select nya</li> </ul>



Kode Test Case	Skenario	Task Sistem Analisis
TC0081	Dapat <i>reset</i> filter (Size, Stroke, dan Colors) sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Terdapat perubahan nama halaman dari "Category" menjadi "Manage Category"</li> <li>• Terdapat perubahan redaksi button + dari "+ Add" → "+ Add Category"</li> <li>• Pada Table <ul style="list-style-type: none"> <li>◦ Terdapat Search Bar</li> <li>◦ Terdapat button dropdown "Bulk Action" yang berisikan <ul style="list-style-type: none"> <li>◦ Delete</li> </ul> </li> <li>◦ Button dropdown "Bulk Action"</li> <li>◦ Terdapat button dropdown filter yang berisikan <ul style="list-style-type: none"> <li>◦ Icon Style</li> <li>◦ Button "Reset Filter"</li> </ul> </li> <li>◦ Terdapat penambahan kolom "MARK" dan "ACTION"</li> <li>◦ Terdapat perubahan di button "Edit" dan "Delete" pada item category</li> </ul> </li> </ul>
TC0082	Dapat mengunduh semua <i>icon</i> per <i>icon style</i> dengan format ZIP sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat mendownload semua icon hanya dengan format ZIP ketika klik "download all icons"</li> </ul>
TC0083	Dapat mencari <i>icon</i> menggunakan filter Pick From Our Icon Collection sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat menampilkan icon dengan kategori yang sama / sesuai dengan kategori yang dipilih</li> </ul>
TC0084	Dapat menemukan <i>icon</i> yang diinginkan menggunakan <i>search bar</i> sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat menampilkan icon dengan size dan stroke yang sudah ditentukan</li> <li>• Di kategori Flat Line hanya menampilkan icon dengan kategori Flat Line</li> </ul>

Kode Test Case	Skenario	Task Sistem Analis
TC0085	Tidak dapat menemukan <i>icon</i> yang tidak sesuai menggunakan <i>search bar</i> sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat menampilkan icon dengan size dan stroke yang sudah ditentukan</li> <li>• Di kategori Flat Line hanya menampilkan icon dengan kategori Flat Line</li> </ul>
TC0086	Dapat <i>mengcopy icon</i> dengan format yang sesuai sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat memasukkan icon ke artboard dengan klik pada icon tersebut</li> </ul>
TC0087	Dapat melakukan <i>logout</i> sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Terdapat pop up yang hanya berisikan <ul style="list-style-type: none"> <li>◦ Logout → ketika di click bisa kembali ke halaman website dengan status tidak login</li> </ul> </li> </ul>
TC0088	Dapat memilih dan mengunduh <i>icon</i> yang sama pada <i>style</i> yang berbeda sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Icon yang terpilih dapat didownload dengan format svg dan zip memiliki nama file dengan format [style_name]_[icon_name]</li> </ul>
TC0096	Dapat mengatur warna dengan hexcode sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat mengatur warna dengan hexcode → bisa dengan klik enter atau klik save</li> <li>• Dapat mengatur warna dengan rgb → bisa dengan klik enter atau klik save</li> </ul>
TC0097	Dapat mengatur warna dengan rgb sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat mengatur warna dengan hexcode → bisa dengan klik enter atau klik save</li> <li>• Dapat mengatur warna dengan rgb → bisa dengan klik enter atau klik save</li> </ul>

Kode Test Case	Skenario	Task Sistem Analisis
TC0112	Dapat menyimpan warna <i>primary</i> , <i>secondary</i> , dan <i>tertiary</i> untuk diterapkan ke semua <i>style</i> sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat menyesuaikan warna element yang memiliki lebih dari satu <i>primary-stroke</i> dan <i>secondary-stroke</i></li> </ul>
TC0118	Dapat melakukan login sebagai member user melalui halaman Register	 <p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Terdapat tulisan "Already have an account?"</li> <li>• Terdapat button Login → mengarahkan user ke halaman login (icon ada di attachment "Login")</li> </ul>
TC0119	Dapat menampilkan <i>icon</i> sesuai dengan <i>icon style</i> dan format ( <i>size</i> , <i>stroke</i> , dan <i>colors</i> ) yang telah diatur sebagai <i>member user</i>	<p><b>Definition of Done</b></p> <ul style="list-style-type: none"> <li>• Dapat menampilkan Icon sesuai dengan format yang telah diatur</li> </ul>

Kode <i>Test Case</i>	Skenario	<i>Task Sistem Analisis</i>
TC0121	Dapat memberi donasi sebagai <i>member user</i>	<b>Definition of Done</b> <ul style="list-style-type: none"> <li>• Tidak perlu di tampilkan di halaman plugin</li> <li>• Dapat membuka halaman paypal pada browser setelah klik 'Buy me a Coffee'</li> </ul>

Pembuatan skenario pengujian *role member user* dikelompokkan ke dalam dari beberapa bagian kategori yaitu *Login, Logout, and Reset Password; Select, Deselect, and Download Icons; Search Icon; Set Colors; dan Basic Functions*. Proses yang termasuk dalam *Basic Function* seperti melakukan donasi, menampilkan *list icon* sesuai dengan filter yang telah diatur, melakukan *copy icon*, dan melakukan *reset filter*. Dalam skenario pengujian *role member user* terdapat 18 skenario uji positif dan 2 skenario uji negatif, sehingga total adalah 20 skenario. Skenario pengujian yang dibuat pada QATouch dapat diekspor ke dalam bentuk tabel yang memuat informasi sebagai berikut:

- a. *Case Code* adalah nomor skenario yang akan diuji, nomor skenario ini berbeda satu dengan yang lainnya sehingga dapat dimaknai sebagai ID dari skenario uji.
- b. *Category* adalah kelompok bagian dari keseluruhan skenario pengujian *role member user*.
- c. *Description* adalah deskripsi dari setiap skenario kasus uji.
- d. *Steps* adalah langkah-langkah pengujian yang harus dilakukan oleh penguji.
- e. *Expected Result* adalah hasil yang diharapkan dari proses pengujian yang telah selesai.
- f. *Actual Result* adalah hasil yang sebenarnya dari proses pengujian yang telah selesai.
- g. *Status* adalah keterangan akhir setelah proses pengujian selesai. Umumnya berupa *Passed* atau *Failed*.

Skenario pengujian *role member user* dapat dilihat pada Tabel 3.3.

Tabel 3.3. Hasil pembuatan skenario pengujian

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0076	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan <i>reset password</i>	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login yang berada di kanan atas halaman Iconhub</li> <li>3. Klik <i>text</i> "Forgot Password"</li> <li>4. <i>Input</i> email yang telah terdaftar</li> <li>5. Klik <i>button</i> Send Link Reset Password</li> <li>6. Klik <i>button</i> "Reset Password" pada email yang telah dikirim</li> <li>7. <i>Input password</i> baru dan konfirmasinya</li> <li>8. Klik <i>button</i> "Reset Password"</li> </ol>	<p>Tampil halaman untuk <i>request reset password</i></p> <p>Tampil <i>alert</i> "We have emailed your password reset link"</p> <p>Tampil halaman Reset Password</p> <p>Tampil halaman Login</p>	-	<i>Passed/Failed</i>
TC0077	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat memilih satu atau banyak <i>icon</i> per <i>icon style</i>	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih Icon Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih</li> <li>7. Klik <i>button</i> Select</li> </ol>	Dapat memilih <i>icon</i> yang diinginkan	-	<i>Passed/Failed</i>
TC0078	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat mengunduh satu atau beberapa <i>icon</i>	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> email dan password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih Icon Style, Size, Stroke dan</li> </ol>	<p>Terpilihnya <i>icon</i></p> <p>Tampil pop up untuk memilih format file yang ingin diunduh</p> <p>Dapat mengunduh <i>icon-icon</i> yang dipilih sesuai format yang dipilih</p>	-	<i>Passed/Failed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button</i> Select 8. Klik <i>button</i> Download ... <i>Icons</i> 9. Klik <i>button</i> ZIP atau SVG sesuai yang diinginkan			
TC0079	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat <i>deselect</i> satu atau banyak <i>icon</i> per <i>icon style</i>	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password 4. Klik <i>button</i> Login 5. Pilih Icon Style, Size, Stroke dan Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button</i> Select 8. <i>Hover mouse</i> ke arah <i>icon</i> yang telah dipilih sebelumnya 9. Klik <i>button</i> Deselect	Dapat memilih <i>icon</i> Dapat <i>deselect</i> pilihan <i>icon</i>	-	<i>Passed/Failed</i>
TC0080	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat <i>deselect</i> seluruh <i>icon</i> yang dipilih	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih Icon Style, Size, Stroke dan Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih	Dapat <i>deselect</i> semua <i>icon</i> yang telah dipilih sebelumnya	-	<i>Passed/Failed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			7. Klik <i>button</i> Select 8. Klik <i>button</i> Deselect All			
TC0081	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat filter reset (Icon Style, Stroke, Colors) dan	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih Icon Style, Size, Stroke dan Colors yang sesuai 6. Klik Reset	Tampil berbagai jenis <i>icon</i> dengan pengaturan semula	-	<i>Passed/Failed</i>
TC0082	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat mengunduh semua <i>icon</i> per Icon Style dengan format ZIP	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih Icon Style, Size, Stroke dan Colors yang sesuai 6. Klik <i>button</i> Download All <i>Icons</i>	Dapat mengunduh semua <i>icon</i> yang ada pada Icon Style yang telah dipilih dengan format ZIP	-	<i>Passed/Failed</i>
TC0083	<i>Search Icon</i>	Sebagai <i>member user</i> dapat mencari <i>icon</i> menggunakan filter Pick From Our <i>Icon</i> Collection	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password 4. Klik <i>button</i> Login 5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai 6. Pilih filter dari dropdown Pick From Our <i>Icon</i> Collection	Tampil berbagai jenis <i>icon</i> dengan filter yang telah diterapkan	-	<i>Passed/Failed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0084	<i>Search Icon</i>	Sebagai <i>member user</i> dapat menemukan <i>icon</i> yang diinginkan menggunakan Search Bar	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Inputkan</i> kata kunci <i>icon</i> yang ingin dicari</li> </ol>	Dapat menampilkan <i>icon</i> sesuai dengan kata kunci yang dicari	-	<i>Passed/Failed</i>
TC0085	<i>Search Icon</i>	Sebagai <i>member user</i> tidak dapat menemukan <i>icon</i> yang tidak sesuai menggunakan Search Bar	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Inputkan</i> kata kunci yang tidak sesuai</li> </ol>	Tampil tulisan "Sorry, <i>Icon</i> Not Found" pada halaman <i>icon</i>	-	<i>Passed/Failed</i>
TC0086	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat mengcopy <i>icon</i> dengan format yang sesuai	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih</li> <li>7. Klik <i>button</i> Copy SVG</li> </ol>	Dapat mengcopy <i>icon</i> dengan format yang sesuai	-	<i>Passed/Failed</i>
TC0087	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan Logout	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login yang berada di kanan</li> </ol>	Tampil <i>pop up</i> Logout Dapat kembali ke halaman Login	-	<i>Passed/Failed</i>



<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			atas halaman Iconhub 3. <i>Input</i> Email dan Password yang telah terdaftar 4. Klik <i>button user profile</i> pada kanan atas 5. Klik <i>pop up Logout</i>			
TC0088	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat memilih dan mengunduh <i>icon</i> yang sama pada <i>style</i> yang berbeda	1. Akses halaman Iconhub 2. Klik <i>button Login</i> 3. <i>Input</i> Email dan Password 4. Klik <i>button Login</i> 5. Pilih salah satu <i>Icon Style, Size, Stroke dan Colors</i> 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button Select</i> 8. Pilih <i>Icon Style</i> yang berbeda 9. <i>Hover mouse</i> ke arah <i>icon</i> dengan nama yang sama dengan yang sebelumnya 10. Klik <i>button Select</i> 11. Klik <i>button Download ... Icons</i> 12. Klik <i>button ZIP atau SVG</i> sesuai yang diinginkan	Tampil berbagai jenis <i>icon</i> dengan <i>style</i> yang berbeda dengan yang dipilih sebelumnya Dapat memilih <i>icon</i> Tampil <i>pop up</i> untuk memilih format file yang ingin diunduh Dapat mengunduh <i>icon-icon</i> yang dipilih sesuai format yang dipilih walaupun dengan nama yang sama	-	<i>Passed/Failed</i>
TC0096	<i>Set Colors</i>	Sebagai <i>member user</i> dapat mengatur warna dengan hexcode	1. Akses halaman Iconhub 2. Klik <i>button Login</i> 3. <i>Input</i> Email dan Password 4. Klik <i>button Login</i> 5. Klik Color box 6. Klik <i>button HEXA</i>	Tampil pengaturan warna Tampil <i>field</i> untuk memasukkan hexcode Tampil warna yang sesuai dengan hexcode yang diinputkan	-	<i>Passed/Failed</i>

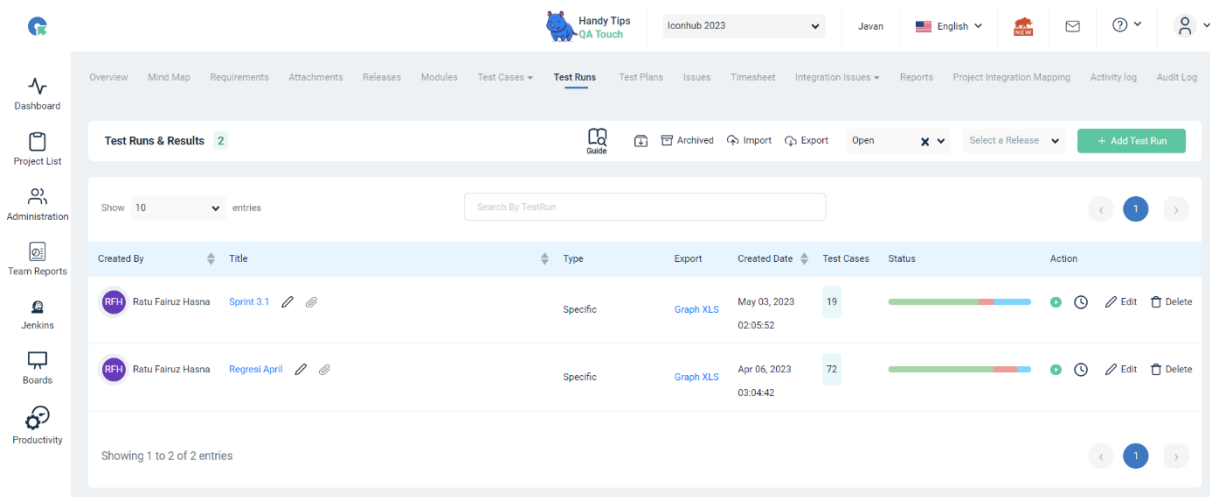
<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			7. <i>Input</i> hexcode yang diinginkan			
TC0097	<i>Set Colors</i>	Sebagai <i>member user</i> dapat mengatur warna dengan RGB	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik Color box</li> <li>6. Klik <i>button</i> RGBA</li> <li>7. <i>Input</i> kode rgb yang diinginkan</li> </ol>	<p>Tampil pengaturan warna</p> <p>Tampil <i>field</i> untuk memasukkan kode warna RGB</p> <p>Tampil warna yang sesuai dengan kode RGB yang diinputkan</p>	-	<i>Passed/Failed</i>
TC0112	<i>Set Colors</i>	Sebagai <i>member user</i> dapat menyimpan warna <i>primary</i> , <i>secondary</i> , dan <i>tertiary</i> untuk diterapkan ke semua style	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>Icon Style</i> "Multi Color"</li> <li>6. Klik Color Box <i>Primary</i></li> <li>7. Pilih warna yang diinginkan</li> <li>8. Klik <i>button</i> Save</li> <li>9. Klik Color Box <i>Secondary</i></li> <li>10. Pilih warna yang diinginkan</li> <li>11. Klik <i>button</i> Save</li> <li>12. Klik Color Box <i>Tertiary</i></li> <li>13. Pilih warna yang diinginkan</li> <li>14. Klik <i>button</i> Save</li> </ol>	<p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>primary</i></p> <p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>secondary</i></p> <p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>tertiary</i></p>	-	<i>Passed/Failed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0116	<i>Basic Functions</i>	Sebagai <i>member user</i> tidak dapat melakukan registrasi dengan email yang telah terdaftar	1. Akses halaman Iconhub 2. Klik <i>button</i> login yang berada di kanan atas halaman Iconhub 3. Klik <i>button</i> Register 4. <i>Input</i> Email yang sudah terdaftar 5. Klik <i>button</i> Register	Tampil <i>alert</i> "The email has already been taken."	-	<i>Passed/Failed</i>
TC0118	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan Login melalui halaman Register	1. Akses halaman Iconhub 2. Klik <i>button</i> login yang berada di kanan atas halaman Iconhub 3. Klik <i>button</i> register 4. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password yang telah terdaftar 5. Klik <i>button</i> Login	Tampil halaman Login Tampil halaman Register Tampil halaman Login Tampil <i>pop up</i> berhasil login dan <i>home page</i> sebagai <i>Member user</i>	-	<i>Passed/Failed</i>
TC0119	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat menampilkan <i>icon</i> sesuai dengan <i>icon</i> style dan format (size, stroke, dan colors) yang telah diatur	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password 4. Klik <i>button</i> Login 5. Klik <i>icon</i> style yang diinginkan 6. Klik Size yang diinginkan 7. Klik Stroke yang diinginkan 8. Klik <i>color box</i> dan pilih warna yang diinginkan	Tampil jenis <i>icon</i> style yang dipilih Tampil <i>icon</i> sesuai dengan <i>size</i> yang dipilih Tampil <i>icon</i> sesuai dengan <i>stroke</i> yang dipilih Tampil <i>icon</i> dengan warna yang sesuai	-	<i>Passed/Failed</i>
TC0121	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat memberi donasi	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password	Tampil <i>pop up</i> "Support Our Work" Diarahkan ke <i>tab</i> baru halaman Paypal milik Bayu	-	<i>Passed/Failed</i>

Case Code	Category	Description	Steps	Expected Result	Actual Result	Status
			4. Klik <i>button</i> Login 5. Klik <i>button</i> Donate 6. Klik <i>button</i> Donate via Paypal	Winata dengan <i>username</i> @bayuhendra		

### 3.3.3 Pengujian Regresi Secara Manual

Setelah membuat skenario pengujian pada QATouch, maka penulis melakukan pengujian regresi secara manual terlebih dahulu. Bukti pengujian regresi secara manual terdapat pada Gambar 3.3.



Gambar 3.3 Bukti pengujian regresi secara manual pada QATouch

Pada *sprint* ke-1 ini penulis melakukan pengujian regresi dengan judul Regresi April yang terdiri dari 72 skenario pengujian. Pada *bar* Status divisualisasikan dengan tiga warna berbeda. Tiga warna tersebut mengacu pada tiga status pengujian, yang berarti hijau untuk pengujian yang berhasil atau *passed*, merah untuk pengujian yang gagal atau *failed*, dan biru untuk pengujian yang belum diuji atau *untested*. Jika dilihat pada Gambar 3.2 bagian Regresi April, terdapat status *untested* yang disebabkan beberapa fitur pada sistem masih dalam tahap pengembangan sehingga belum dapat dilakukan pengujian terhadap fitur tersebut.

Pada akhir *sprint* ke-1 diadakan *sprint review* untuk membahas pekerjaan yang sudah dan yang belum diselesaikan dalam satu *sprint*. Jika ada pekerjaan yang belum diselesaikan pada *sprint* tersebut maka akan menjadi tambahan pekerjaan pada *sprint* berikutnya. Pada *sprint* ke-1, penulis telah menyelesaikan hampir seluruh pekerjaan dengan tepat waktu kecuali pembuatan dokumen *user guide*, sehingga pembuatan dokumen *user guide* dimasukkan pada *sprint* selanjutnya.

### 3.4 Sprint Ke-2 dan Ke-3

Sprint ke-2 berlangsung pada tanggal 17 April 2023 hingga 28 April 2023, sedangkan sprint ke-3 berlangsung pada tanggal 1 Mei 2023 hingga 12 Mei 2023. Pada *sprint* ke-2 dan ke-3, penulis bertugas untuk melanjutkan pembuatan dokumen *user guide*, melakukan *update* skenario pengujian yang sebelumnya telah dibuat karena *website* Iconhub masih dalam pengembangan, sehingga dapat dipastikan ada penambahan dan/atau pembaruan skenario pengujian, serta penulis juga mendapat tugas baru yaitu membuat pengujian otomatis untuk *website* Iconhub menggunakan Selenium WebDriver dan mengintegrasikannya dengan *third party tool* yang bernama Allure Framework Report.

Seperti yang sudah disebutkan pada penelitian García et al., (2020), bahwa salah satu keuntungan Selenium WebDriver adalah memiliki komunitas yang besar dikarenakan Selenium WebDriver merupakan salah satu alat pengujian otomatis yang populer dikalangan penguji, sehingga ketika ada kesulitan dapat dengan mudah mencari solusinya dan begitu pula dengan Allure Framework Report. Melalui survey yang telah dilakukan oleh García et al., (2020), Allure Framework Report digunakan oleh 11,11% responden dan 62,5% lainnya tidak menggunakan alat pelaporan hasil uji apapun. Penelitian tersebut memperkuat alasan penulis memilih menggunakan Selenium WebDriver serta Allure Framework Report untuk melakukan pengujian regresi *website* Iconhub selain dikarenakan ketentuan dari perusahaan.

Pada Gambar 3.4 terdapat bukti penugasan pada Taiga untuk pembuatan beberapa skrip tes dengan Selenium WebDriver.

**Acceptance Criteria**

1. Terdapat script Selenium untuk test case TC0077
2. Terdapat script Selenium untuk test case TC0078
3. Terdapat script Selenium untuk test case TC0079
4. Terdapat script Selenium untuk test case TC0080
5. Terdapat script Selenium untuk test case TC0082
6. Terdapat script Selenium untuk test case TC0088

**0 Attachments** +

Drop attachments here!

**Tasks** +

#353 Terdapat-script-Selenium-untuk-test-case-TC0077	Edit with S2 Closed	Ratu Fair...
#354 Terdapat-script-Selenium-untuk-test-case-TC0078	Edit with S2 Closed	Ratu Fair...
#355 Terdapat-script-Selenium-untuk-test-case-TC0079	Edit with S2 Closed	Ratu Fair...
#356 Terdapat-script-Selenium-untuk-test-case-TC0080	Edit with S2 Closed	Ratu Fair...
#357 Terdapat-script-Selenium-untuk-test-case-TC0082	Edit with S2 Closed	Ratu Fair...
#358 Terdapat-script-Selenium-untuk-test-case-TC0088	Edit with S2 Closed	Ratu Fair...

Gambar 3.4 Bukti penugasan pada Taiga

### 3.4.1 Instalasi Selenium

Sebelum pembuatan dan eksekusi skrip tes, penulis harus mempersiapkan kerangka kerja Selenium. Pertama penulis harus memastikan Node.js sudah terinstal pada sistem. Kedua, untuk menginisiasi sebuah *project* maka penulis menjalankan `npm init` atau `npm init -y` pada terminal untuk menginisiasi *project* secara *default*. Selanjutnya, penulis dapat melakukan instalasi Selenium dengan menjalankan perintah seperti pada Gambar 3.5 pada terminal perangkat lunak penyunting kode yang digunakan.

```
npm install selenium-webdriver
npm install chromedriver
npm install dotenv
```

Gambar 3.5 Perintah instalasi Selenium

`npm install selenium-webdriver` digunakan untuk menginstal paket atau pustaka dari Selenium WebDriver dalam lingkungan JavaScript atau Node.js. Dengan menginstal paket tersebut, penulis dapat menggunakan Selenium WebDriver untuk mengontrol *browser* web dan melakukan otomatisasi Tindakan-tindakan pada *browser* web tersebut.

`npm install chromedriver` digunakan untuk menginstal Chromedriver, yaitu sebuah server WebDriver yang memungkinkan penggunaan Selenium WebDriver dengan *browser* Google Chrome. Chromedriver dibutuhkan dalam pengujian otomatis yang menggunakan Selenium WebDriver dengan *browser* Google Chrome karena Chromedriver yang akan menjadi perantara untuk dapat berinteraksi dengan *browser* tersebut secara otomatis melalui kode JavaScript atau Node.js. Perlu diperhatikan untuk instalasi Chromedriver harus sama dengan versi *browsersnya*.

`npm install dotenv` digunakan untuk menginstal paket atau pustaka dotenv pada proyek JavaScript atau Node.js. dotenv adalah sebuah pustaka yang memungkinkan pengguna untuk mengelola *environment variables* dalam aplikasi. Dengan menggunakan dotenv pengguna dapat membuat file `.env` dimana dapat menyimpan *environment variables* seperti kunci API, alamat *database*, atau konfigurasi lainnya.

### 3.4.2 Pembuatan Skrip Tes

Skenario pengujian yang telah dibuat sebelumnya menggunakan QATouch menjadi acuan dalam pembuatan skrip tes. Skenario pengujian yang telah dibuat secara manual dituliskan pada perangkat lunak penyunting kode bernama Visual Studio Code atau yang biasa disebut dengan VSCode sehingga menjadi sebuah skrip tes untuk pengujian otomatis.

Pertama kali yang harus dituliskan dalam sebuah skrip tes adalah kode untuk mengimpor fungsi, kelas, dan objek dari pustaka-pustaka yang terdapat pada lingkungan JavaScript atau Node.js seperti pada Gambar 3.6.

```
const { assert } = require('chai');
const { Builder, By, Key, until } = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver');
const loginModule = require('./Login/TC_0075');
```

Gambar 3.6 Kode pengimporan fungsi, kelas dan objek

`const { assert } = require('chai')` digunakan untuk mengimpor fungsi `'assert'` dari pustaka bernama Chai dalam lingkungan JavaScript atau Node.js. Pustaka Chai adalah salah satu pustaka yang populer dalam ekosistem JavaScript. Dengan menggunakan fungsi `'assert'`, penguji dapat memeriksa kondisi yang diharapkan pada pengujian dan memastikan hasilnya sesuai dengan yang diinginkan. Ada banyak cara untuk melakukan validasi atas proses yang dilakukan agar sama dengan yang diinginkan selain dengan fungsi `'assert'`, seperti salah satunya adalah menggunakan fungsi `'expect'`. Fungsi `'expect'`, diimpor dari pustaka Chai dengan kode `var expect = require('chai').expect`. Dengan menggunakan fungsi `'expect'`, penguji dapat membandingkan nilai yang diharapkan dengan nilai aktual yang diperoleh dalam pengujian.

`const { Builder, By, Key, until } = require('selenium-webdriver')` digunakan untuk mengimpor kelas-kelas dan objek-objek yang diperlukan dari pustaka `'selenium-webdriver'` dalam lingkungan JavaScript atau Node.js. Kelas dan objek yang diimpor tersebut memberikan akses ke berbagai fitur dan fungsi yang diperlukan dalam pengujian otomatis menggunakan Selenium WebDriver.

`const loginModule = require('./Login/TC_0075')` digunakan untuk mengimpor file eksternal `TC_0075.js` yang berupa skrip tes untuk melakukan *login* sebagai *Member user*, sehingga tidak perlu menuliskan langkah untuk *login* sebagai *Member user* berulang kali ke dalam tiap skrip tes. Dengan memisahkan fungsionalitas ke dalam modul terpisah penguji dapat mengorganisir dan mengelola kode dengan lebih baik, serta memfasilitasi pemakaian ulang dan pemeliharaan kode yang lebih mudah.

Kemudian setelah mengimpor fungsi, kelas, dan objek yang dibutuhkan dalam skrip maka selanjutnya adalah penulisan inti dari skrip tes ini. Pada skrip dituliskan `'describe'` yang kemudian diikuti oleh blok `'it'`. `'describe'` digunakan untuk mengelompokkan sekumpulan pengujian yang berhubungan secara logis dan blok `'it'` digunakan untuk

mendefinisikan setiap kasus uji yang terkait dengan 'describe' tersebut. Di dalam blok 'it', penguji dapat mengimplementasikan skenario kasus uji yang memeriksa perilaku atau fungsionalitas dari fitur Iconhub. Penulisan 'describe' dan blok 'it' ditunjukkan pada Gambar 3.7.

```
describe('iconhub', function(){
  describe('Web', function(){
    describe('Member user', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0080 - Sebagai member dapat deselect semua icon yang
      dipilih', async function() {

        //Deselect All
        await driver.sleep(2000);
        var ele=
        driver.wait(until.elementLocated(By.xpath("//button[
        contains(text(),'DESELECT ALL')]")));
        await ele.click();
        await driver.sleep(2000);

        //Validasi
        let elements = await
        driver.findElements(By.xpath("/div[contains(@class,'
        shadow-highlight')]"));
        if (elements.length === 0) {
          console.log("Deselect All berhasil, tidak ada icon
          yang terpilih.");
        }

      })

    })

  })

})
```

Gambar 3.7 Penulisan 'describe' dan blok 'it'

Gambar-gambar berikutnya menunjukkan penulisan skrip kode yang sudah penulis buat berdasarkan skenario pengujian sebelumnya. Dari 20 skenario pengujian *role member user* terbagi ke dalam 13 skrip tes, beberapa skenario pengujian tergabung ke dalam satu *file* skrip tes yang sama dikarenakan langkah pengujian yang saling berhubungan. Pada kasus ini TC0078 dan TC0080 menjadi satu *file* skrip tes dengan TC0077, kemudian TC0084 dengan TC0085, serta TC0096 dengan TC0097.



```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        driver = await new
Builder().forBrowser('chrome').build();
        vars = {}
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0076 - Sebagai member dapat melakukan reset
password', async function() {
        await driver.get("https://iconhub.merapi.javan.id/");
        await driver.manage().window().maximize();

        //login
        var ele=
driver.wait(until.elementLocated(By.linkText('Login')));
        await ele.click();

        await driver.findElement(By.linkText('Forgot
Password?')).click();

        var ele=
driver.wait(until.elementLocated(By.id('email')));
        await ele.sendKeys('member@iconhub.io');
        await driver.sleep(2000);

        await
driver.findElement(By.xpath('/html/body/div[1]/div/div/form/div[2]/butto
n')).click();
        await driver.sleep(2000);

        //validasi
        await
driver.findElements(By.xpath("//p[contains(text(),'We have emailed your
password reset link!')]"));
      })
    })
  })
})

```

Gambar 3.8 Skrip tes TC0076

Pada gambar 3.8 menunjukkan skrip tes TC0076 yang merupakan skenario pengujian untuk dapat melakukan *reset password* sebagai *member user*. Ketika *member user* lupa *password* akun Iconhubnya, maka *member user* dapat klik *link text* yang berbunyi “Forgot Password?” yang terdapat pada halaman Login. Maka selanjutnya *member user* akan diperintahkan untuk memasukkan alamat email akun Iconhub mereka agar sistem dapat

mengirimkan *link* untuk *reset password* mereka melalui email yang terdaftar. Ketika *member user* telah mengisi alamat email mereka dan klik *button* yang berbunyi “Send Link Reset Password” maka akan muncul notifikasi berbunyi “We have emailed your password reset link!”.

```

const { assert } = require('chai');
const { Builder, By, Key, until } = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function() {
  describe('Web', function() {
    describe('Member User', function() {

      before(async function() {
        await loginModule.login();
      })
      after(async function() {
        await driver.quit();
      })
      it('TC0077 - Sebagai member dapat memilih satu atau
beberapa icon per icon style', async function() {

        //pilih icon style
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[3]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[2]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        await driver.findElement(By.xpath('//*[@id="icon-
list"]/div[1]/div[10]/div[1]/div[2]/button[1]')).click();
        await driver.sleep(2000);

        //pilih icon style
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[6]/li/
button')).click();
        await driver.sleep(2000);

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[4]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        // //validasi
        await
driver.findElement(By.xpath("//div[contains(@class, 'shadow-
highlight')]"));
      })
    }
  }
}

```

```

        it('TC0078 - Sebagai member dapat download satu atau
beberapa icon yang dipilih', async function() {

            //Download satu atau beberapa icon
            var ele=
driver.wait(until.elementLocated(By.xpath("//button[contains(text (),
'Download')]")));
            await ele.click();
            await driver.sleep(2000);

            //pilih format file
            var ele=
driver.wait(until.elementLocated(By.xpath('/html/body/div[1]/div/div[3]/
div[2]/div/div[2]/div[3]/button[1]')));
            await ele.click();
            await driver.sleep(2000);

            //validasi download
            await
driver.findElement(By.xpath('//div[contains(text(), "downloading
Iconhub!')]'));

            //close pop up download
            await
driver.findElement(By.xpath('/html/body/div[1]/div/div[3]/div[4]/div/div
[1]')).click();
            await driver.sleep(2000);
        })

        it('TC0080 - Sebagai member dapat deselect semua icon
yang dipilih', async function() {

            //Deselect All
            await driver.sleep(2000);
            var ele=
driver.wait(until.elementLocated(By.xpath("//button[contains(text (), 'DES
ELECT ALL')]")));
            await ele.click();
            await driver.sleep(2000);

            //Validasi
            let elements = await
driver.findElements(By.xpath("/div[contains(@class, 'shadow-
highlight')]"));
            if (elements.length === 0) {
                console.log("Deselect All berhasil, tidak ada
icon yang terpilih.");
            }

        })

    })
})

```

Gambar 3.9 Skrip tes TC0077, TC0078, dan TC0080

Pada gambar 3.9 menunjukkan skrip tes TC0077 yang merupakan skenario mengenai *member user* dapat memilih satu atau beberapa *icon* dengan *icon style* yang berbeda, TC0078 mengenai *member user* dapat mengunggah *icon-icon* yang sudah dipilih sebelumnya, sedangkan TC0080 mengenai *member user* dapat melakukan *deselect* semua *icon* yang sudah dipilih. Ketiga skenario pengujian tersebut berhubungan satu dengan yang lainnya maka dari itu untuk menghindari penulisan langkah yang berulang-ulang maka ketiga skenario pengujian ini dapat disatukan ke dalam satu *file* yang sama hanya saja dengan blok `it` yang berbeda.

```

const { assert } = require('chai');
const { Builder, By, Key, until } = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver');
const loginModule = require('./Login/TC_0075');

describe('iconhub', function() {
  describe('Web', function() {
    describe('Member User', function() {

      before(async function() {
        await loginModule.login();
      })
      after(async function() {
        await driver.quit();
      })
      it('TC0079 Sebagai member dapat deselect satu atau
beberapa icon yang dipilih', async function() {

        //pilih icon style
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[3]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[2]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        await driver.findElement(By.xpath('//*[@id="icon-
list"]/div[1]/div[10]/div[1]/div[2]/button[1]')).click();
        await driver.sleep(2000);

        //pilih icon style
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[6]/li/
button')).click();
        await driver.sleep(2000);

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[4]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        // //validasi
        await

```

```

        //pilih icon style
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[3]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //klik deselect
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[2]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        await driver.findElement(By.xpath('//*[@id="icon-
list"]/div[1]/div[10]/div[1]/div[2]/button[1]')).click();
        await driver.sleep(2000);

        //Validasi
        let mc = await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[6]/li/
button/div/span')).getText();
        expect(mc).to.equal('1');

        let lc = await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[3]/li/
button/div/span')).getText();
        expect(lc).to.equal('');
    })
})
})
})

```

Gambar 3.10 Skrip tes TC0079

Pada Gambar 3.10 menunjukkan skrip tes TC0079 yang merupakan skenario pengujian mengenai sebagai *member user* dapat melakukan *deselect* satu atau beberapa *icon* yang sudah dipilih sebelumnya. Ketika *member user* telah memilih *icon*, *member user* dapat *deselect* pilihan tersebut dengan klik *button* yang berbunyi “Deselect” pada *icon* yang terpilih sebelumnya. Pada website Iconhub, *icon* yang terpilih akan ditandai dengan *box icon* tersebut *terhighlight* berwarna ungu tua sedangkan jika *deselect* maka *box icon* tersebut akan kembali seperti semula.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0081 - Sebagai member dapat mereset Size, Stroke,
dan Color', async function() {

        //pick size 64
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/div/div[1]/div[2]/div[5]/div')));
        await ele.click();
        await driver.sleep(2000);
        //pick stroke 3
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[2]/di
v[2]/div[6]/div')).click();
        await driver.sleep(2000);
        //scroll down biar ga ketutupan php debugger
        let element = await
driver.findElement(By.className('sticky top-0 md:h-screen md:overflow-y-
auto px-5 py-8'));
        await driver.executeScript('arguments[0].scrollTop +=
700;', element);

        //pick color
        await driver.findElement(By.className('pcc-
button')).click();
        await driver.sleep(2000);

        const field = driver.findElement(By.className('pcc-
result'));

        field.click();
        field.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await driver.findElement(By.className('pcc-
result')).sendKeys('#D51313');
        await driver.sleep(2000);
        await driver.findElement(By.className('pcc-
save')).click();
        await driver.sleep(2000);

```

```

        //klik reset button
        await driver.findElement(By.className('block text-center text-xs text-purple-700 hover:bg-gray-100 active:bg-gray-300 uppercase rounded font-semibold py-3 focus:shadow-none focus:outline-none')).click();
        await driver.sleep(2000);

        //validasi
        await driver.findElements(By.className('text-xs font-semibold text-center align-middle rounded border border-gray-300 w-8 p-3/2 cursor-pointer btn-icon-size text-white bg-purple-700 border-purple-700 shadow-primary'));
        await driver.findElements(By.className('text-xs font-semibold text-center align-middle rounded border border-gray-300 py-3/2 w-8 cursor-pointer btn-icon-stroke text-white bg-purple-700 border-purple-700 shadow-primary'));
    })
    })
})

```

Gambar 3.11 Skenario tes TC0081

Pada gambar 3.11 menunjukkan skrip tes TC0081 yang merupakan skenario pengujian mengenai member user dapat melakukan *reset size*, *stroke*, dan *color* yang sudah diatur sebelumnya. Pada *website* Iconhub, *member user* dapat mengatur ukuran *icon*, ketebalan *stroke icon*, dan juga warna dari *icon* sesuai dengan kebutuhan pengguna. *Member user* juga dapat melakukan *reset* atau kembali ke pengaturan *default* yang telah ditetapkan oleh *website* Iconhub itu sendiri dengan cara klik *button* yang bertuliskan “Reset”. Ketika pengaturan *icon* telah *direset*, maka ukuran *icon* akan kembali ke ukuran 48px dan *stroke icon* akan kembali menjadi 2px.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0082 - Sebagai member dapat mendownload seluruh icon per icon style', async function() {

```

```

        //pick icon style
        var ele=
driver.wait (until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[2]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //download all
        await
driver.findElement (By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[3]/bu
tton[2]')).click();
        await driver.sleep(2000);

        //validasi download
        await

```

Gambar 3.12 Skrip tes TC0082

Pada gambar 3.12 menunjukkan skrip tes TC0082 yang merupakan skenario pengujian mengenai *member user* dapat mengunduh keseluruhan *icon* yang ada pada *icon style* yang dipilih dengan klik button “Download All”. Ketika *member user* klik *button* “Download All”, maka seluruh *icon* pada *icon style* yang dipilih akan terunduh ke dalam satu *folder* dengan format ZIP.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0083 - Sebagai member dapat mencari icon dari
dropdown Category', async function() {

        //pick from category
        var ele=
driver.wait (until.elementLocated(By.id('dropdownCategory')));
        await ele.click();
        await driver.sleep(2000);
        //category: Brands
        await
driver.findElement (By.xpath('//*[@id="dropdownCategory"]/option[6]')).cl
ick();

        await driver.sleep(2000);

```



```

        //validasi ada icon Discord di kategori Brands
        await driver.findElement(By.id('icons-discord'));
    })
})
})
})

```

Gambar 3.13 Skrip tes TC0083

Pada gambar 3.13 menunjukkan skrip tes TC0083 yang merupakan skenario pengujian mengenai member user dapat mencari *icon* yang terdapat pada kategori tertentu. *Member user* memilih kategori sesuai yang dibutuhkan kemudian dapat memilih *icon* tertentu yang disukai. Pada skrip tes pengujian dilakukan dengan menemukan *icon* Discord pada kategori Brands.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0084 - Sebagai member dapat mencari icon pada
search bar', async function() {

        //search bar
        var ele=
driver.wait(until.elementLocated(By.id('searchInput')));
        await ele.sendKeys('shipping');
        await driver.sleep(2000);

        //validasi
        await driver.findElement(By.id('icons-car-
shipping'));
      })

      it('TC0085 - Sebagai member tidak dapat menemukan icon
yang tidak sesuai pada search bar', async function() {

        //search bar
        var ele=
driver.wait(until.elementLocated(By.id('searchInput')));
        await ele.sendKeys('zzz');
        await driver.sleep(2000);

```

```

        //validasi
        await
driver.findElement(By.xpath('//div[contains(@class, "icon-not-
found")]'));
    })
    })
    })
})

```

Gambar 3.14 Skrip tes TC0084 dan TC0085

Pada gambar 3.14 menunjukkan skrip tes TC0084 yang merupakan skenario pengujian mengenai *member user* dapat mencari *icon* dengan memasukan kata kunci yang sesuai menggunakan *search bar*, sehingga jika *icon* yang dicari terdapat pada sistem maka akan memunculkan *icon* yang dicari tersebut. Sedangkan pada TC0085 adalah skenario pengujian negatif ketika *icon* yang dicari tidak terdapat pada sistem maka akan memunculkan informasi berbunyi “Sorry, icon not found.”

```

const { assert } = require('chai');
const { Builder, By, Key, until } = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver');
const loginModule = require('./Login/TC_0075');

describe('iconhub', function() {
  describe('Web', function() {
    describe('Member User', function() {

      before(async function() {
        await loginModule.login();
      })
      after(async function() {
        await driver.quit();
      })
      it('TC0086 - Sebagai member dapat mengcopy icon', async
function() {

        //pilih icon style
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[3]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //klik Copy SVG
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[2]/div[1]/div[2]/button[2]')));
        await ele.click();
        await driver.sleep(500);

```

```

        // //validasi
        let copy = await
driver.findElement(By.xpath('//*[@id="icon-
list"]/div[1]/div[2]/div[1]/div[2]/button[2]')).getText();
        expect(copy).to.equal('Copied!');
    })
})
})
})

```

Gambar 3.15 Skrip tes TC0086

Pada gambar 3.15 menunjukkan skrip tes TC0086 yang merupakan skenario pengujian mengenai *member user* dapat menjalankan fungsi *copy icon* sehingga *icon* yang *dicopy* dapat dimasukkan ke dalam *artboard*. Jika *member user* klik *button copy* yang ada pada *icon*, maka *button copy* tersebut akan berubah menjadi “Copied!” yang berarti *icon* tersebut telah berhasil *dicopy*.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){

      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0087 - Sebagai member dapat logout', async
function() {

        //klik profile
        var ele=
driver.wait(until.elementLocated(By.xpath('/html/body/div[1]/div/div[1]/
button')));

        await ele.click();
        await driver.sleep(2000);

        await
driver.findElement(By.xpath('/html/body/div[1]/div/div[1]/div/ul/li/a'))
.click();

        await driver.sleep(2000);

```

```

// //validasi kembali ke halaman login
var ele=
driver.wait(until.elementLocated(By.xpath('//span[contains(text(),
>Login"')]')));
    })
    })
    })
})

```

Gambar 3.16 Skrip tes TC0087

Pada gambar 3.16 menunjukkan skrip tes TC0087 yang merupakan skenario pengujian mengenai *member user* dapat keluar dari akun *member user*nya atau yang disebut juga dengan melakukan *logout*. Fungsi *logout* terdapat ketika pengguna klik *profile* pada bagian kanan atas kemudian klik *button Logout*. Jika *member user logout*, maka akan kembali ke halaman Login.

```

const { assert } = require('chai');
const { Builder, By, Key, until } = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver');
const loginModule = require('./Login/TC_0075');

describe('iconhub', function() {
  describe('Web', function() {
    describe('Member User', function() {

      before(async function() {
        await loginModule.login();
      })
      after(async function() {
        await driver.quit();
      })
      it('TC0088 - Sebagai member dapat mendownload icon yang
sama pada style berbeda', async function() {

        //pilih icon style
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[4]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[3]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        //pilih icon style
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[5]/li/
button')).click();
        await driver.sleep(2000);

```

```

        //klik select
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="icon-
list"]/div[1]/div[3]/div[1]/div[2]/button[1]')));
        await ele.click();
        await driver.sleep(2000);

        // //validasi
        await
driver.findElement(By.xpath("//div[contains(@class,'shadow-
highlight')]"));
        //Download satu atau beberapa icon
        var ele=
driver.wait(until.elementLocated(By.xpath("//button[contains(text (),
'Download')]")));
        await ele.click();
        await driver.sleep(2000);

        //pilih format file
        var ele=
driver.wait(until.elementLocated(By.xpath('/html/body/div[1]/div/div[3]/
div[2]/div/div[2]/div[3]/button[1]')));
        await ele.click();
        await driver.sleep(2000);
    //}

    //validasi download
    await
driver.findElement(By.xpath('//div[contains(text(), "downloading
Iconhub!")]'));
    })
    })
    })
    })
})

```

Gambar 3.16 Skrip tes TC0088

Pada gambar 3.16 menunjukkan skrip tes TC0088 yang merupakan skenario pengujian mengenai *member user* dapat mengunduh *icon* yang sama pada *icon style* yang berbeda. Pengujian dilakukan dengan memilih *icon style* terlebih dahulu kemudian memilih *icon* yang terdapat pada *icon style* tersebut, setelah itu memilih *icon style* yang lain dan memilih *icon* dengan nama yang sama dengan *icon* yang sebelumnya. Langkah terakhir yaitu klik *button Download .. icons* maka *icon* yang dipilih akan langsung terunduh.

```

const { assert } = require('chai');
const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0096 - Sebagai member dapat mengatur warna dengan
hex code', async function() {

        const scroll = await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[4]/a[
2]'));
        await
driver.executeScript("arguments[0].scrollIntoView()", scroll);

        //pick color hex
        await driver.findElement(By.className('pcr-
button')).click();
        await driver.sleep(2000);

        const field = driver.findElement(By.className('pcr-
result'));
        field.click();
        field.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await driver.findElement(By.className('pcr-
result')).sendKeys('#D51313');
        await driver.sleep(2000);
        await driver.findElement(By.className('pcr-
save')).click();
        await driver.sleep(2000);

        //validasi warna icon
        await
driver.findElement(By.xpath("//div[contains(@class,'icons-
wrapper')]//*[contains(@style,'rgb(213, 19, 19)')]"));

      })

```

```

    it('TC0097 - Sebagai member dapat mengatur warna dengan rgb code',
    async function() {

        const scroll = await
    driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[4]/a[
    2]'));
        await
    driver.executeScript("arguments[0].scrollIntoView()", scroll);

        //pick color rgb
        await driver.findElement(By.className('pcc-
    button')).click();
        await driver.sleep(2000);

        await
    driver.findElement(By.xpath('/html/body/div[5]/div[3]/input[3]')).click(
    );
        await driver.sleep(2000);

        const field = driver.findElement(By.className('pcc-
    result'));
        field.click();
        field.sendKeys(Key.chord(Key.CONTROL, 'a')); //
    select all text
        field.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
    delete all text

        await driver.findElement(By.className('pcc-
    result')).sendKeys('rgba(222, 120, 120, 1)');
        await driver.sleep(2000);
        await driver.findElement(By.className('pcc-
    save')).click();
        await driver.sleep(2000);

        //validasi warna icon
        await
    driver.findElement(By.xpath("//div[contains(@class, 'icons-
    wrapper')]//*[contains(@style, 'rgb(222, 120, 120)']"));
        })
    })
})

```

Gambar 3.17 Skenario tes TC0096 dan TC0097

Pada gambar 3.17 menunjukkan skrip tes TC0096 yang merupakan skenario pengujian mengenai *member user* dapat mengatur warna *icon* menggunakan *HEX code* sedangkan TC0097 mengenai *member user* dapat mengatur warna *icon* menggunakan *RGB code*. Fungsi *hex* dan *rgb* ini dapat ditemukan ketika klik *color box* yang ada pada *side bar* sebelah kiri.

```

const { assert } = require('chai');
const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0112 - Sebagai member dapat menerapkan warna pada
icon style', async function() {

        //pilih icon style Multi Color
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[6]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        const scroll = await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[4]/a[
2]'));
        await
driver.executeScript("arguments[0].scrollIntoView()", scroll);

        //pick color primary
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[3]/di
v[2]/div/button')).click();
        await driver.sleep(2000);

        const field1 = driver.findElement(By.className('pccr-
result'));
        field1.click();
        field1.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field1.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await driver.findElement(By.className('pccr-
result')).sendKeys('#BF0E77');
        await driver.sleep(2000);
        await driver.findElement(By.className('pccr-
save')).click();
        await driver.sleep(2000);

```



```

    //pick color Secondary
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[3]/div[3]/div/button')).click();
        await driver.sleep(2000);

        const field2 =
driver.findElement(By.xpath('/html/body/div[6]/div[3]/input[1]'));
        field2.click();
        field2.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field2.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await
driver.findElement(By.xpath('/html/body/div[6]/div[3]/input[1]')).sendKeys('#FAD600');
        await driver.sleep(2000);
        await
driver.findElement(By.xpath('/html/body/div[6]/div[3]/input[7]')).click();
        await driver.sleep(2000);

    //pick color Tertiary
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[3]/div[4]/div/button')).click();
        await driver.sleep(2000);

        const field3 =
driver.findElement(By.xpath('/html/body/div[7]/div[3]/input[1]'));
        field3.click();
        field3.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field3.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await
driver.findElement(By.xpath('/html/body/div[7]/div[3]/input[1]')).sendKeys('#5B9EF3');
        await driver.sleep(2000);
        await
driver.findElement(By.xpath('/html/body/div[7]/div[3]/input[7]')).click();
        await driver.sleep(2000);

    //pilih icon style Flat Line
    var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]/ul/div[5]/li/button')));
        await ele.click();
        await driver.sleep(2000);

```

```

        //validasi warna icon primary
        await
driver.findElement(By.xpath("//div[contains(@class,'icons-
wrapper')]//*[contains(@style,'rgb(191, 14, 119)')]"));
        //validasi warna icon secondary
        await
driver.findElement(By.xpath("//div[contains(@class,'icons-
wrapper')]//*[contains(@style,'rgb(250, 214, 0)')]"));

        //pilih icon style Flat Color
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/ul/div[4]/li/button')));
        await ele.click();
        await driver.sleep(2000);

        const primary =
driver.wait(until.elementLocated(By.id('primary'))); // mengambil elemen
SVG
        const style1 = await primary.getAttribute('style');
// mengambil nilai atribut width
        assert.equal(style1, 'fill: rgb(191, 14, 119);'); //
membandingkan nilai atribut dengan nilai yang diharapkan

        const secondary =
driver.wait(until.elementLocated(By.id('secondary')));
        const style2 = await secondary.getAttribute('style');
        assert.equal(style2, 'fill: rgb(250, 214, 0);');
    })
    })
})

```

Gambar 3.18 Skrip tes TC0112

Pada gambar 3.18 menunjukkan skrip tes TC0112 yang merupakan skenario pengujian mengenai *member user* dapat menerapkan seluruh warna yaitu *primary*, *secondary*, dan *tertiary* yang telah diatur pada *icon style* Multi Colors ke dalam seluruh *icon style* kecuali *icon style* Line dan Glyph. Pengujian yang dilakukan yaitu dengan mengatur seluruh warna pada *icon style* Multi Colors dan ketika klik *icon style* yang lain maka warnanya akan mengikuti warna yang telah diatur pada *icon style* Multi Colors.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        driver = await new
Builder().forBrowser('chrome').build();
        vars = {}
      })
      after(async function(){
        await driver.quit();
      })
    })
  })
})

```

```

        it('TC0116 - Sebagai member tidak dapat register dengan
email terdaftar', async function() {
            await driver.get("https://iconhub.merapi.javan.id/");
            await driver.manage().window().maximize();

            //login
            var ele=
driver.wait(until.elementLocated(By.linkText('Login')));
            await ele.click();
            await
driver.findElement(By.xpath('//span[contains(text(),
"Register")]')).click();

            //input data dengan email yang telah terdaftar
            var ele=
driver.wait(until.elementLocated(By.id('email')));
            await ele.sendKeys('member@iconhub.io');
            await
driver.findElement(By.id('password')).sendKeys('asdf1234');
            await
driver.findElement(By.id('fullname')).sendKeys('member');
            await driver.sleep(2000);

            await
driver.findElement(By.xpath('//span[contains(text(),
"Register")]')).click();

            //validasi
            await
driver.findElements(By.xpath("/html/body/div[1]/div/div/form/div[2]"));
        })
    })
})

```

Gambar 3.19 Skrip tes TC0116

Pada gambar 3.19 menunjukkan skrip tes TC0116 yang merupakan skenario pengujian mengenai *member user* tidak dapat melakukan registrasi dengan email yang telah terdaftar. Jika *member user* melakukan *register* dengan email yang telah terdaftar sebelumnya, maka akan ada peringatan berbunyi “The email has already been taken.”

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')

describe('iconhub', function(){
    describe('Web', function(){
        describe('Member User', function(){
            before(async function(){
                driver = await new
Builder().forBrowser('chrome').build();
                vars = {}

```

```

    })
    after(async function(){
        await driver.quit();
    })
    it('TC0118 - Sebagai member dapat melakukan login
melalauai halaman register', async function() {
        await driver.get("https://iconhub.merapi.javan.id/");
        await driver.manage().window().maximize();

        //login
        var ele=
driver.wait(until.elementLocated(By.linkText('Login')));
        await ele.click();

        await
driver.findElement(By.linkText('Register')).click();
        await driver.sleep(2000);

        await
driver.findElement(By.linkText('Login')).click();
        await driver.sleep(2000);

        await
driver.findElement(By.id('email')).sendKeys('member@iconhub.io');
        await
driver.findElement(By.id('password')).sendKeys('asdf1234');
        await driver.sleep(2000);

        await
driver.findElement(By.xpath('/html/body/div[1]/div/div/form/div[3]/butto
n')).click();
        await driver.sleep(2000);

        //validasi
        await
driver.wait(until.elementLocated(By.xpath('//span[contains(text(), "User
Member")]')));
    })
})
})
})

```

Gambar 3.20 Skrip tes TC0118

Pada gambar 3.20 menunjukkan skrip tes TC0118 yang merupakan skenario pengujian mengenai *member user* dapat melakukan *login* melalui halaman Registrasi. Pada halaman Registrasi terdapat *button login* yang dapat digunakan ketika sudah memiliki akun yang terdaftar, sehingga pengguna tidak perlu melakukan registrasi ulang. Pengujian dilakukan dengan masuk ke halaman Registrasi dari halaman Login, kemudian klik *button login*, sehingga akan kembali ke halaman Login.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const assert = require('assert');
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0119 - Sebagai member dapat menampilkan icon sesuai
dengan format', async function() {

        //pick size 64
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/div/div[1]/div[2]/div[5]/div')));
        await ele.click();
        await driver.sleep(2000);
        //pick stroke 3
        await
driver.findElement(By.xpath('//*[@id="sidebar"]/div/div[3]/div/div[2]/di
v[2]/div[6]/div')).click();
        await driver.sleep(2000);
        //scroll down biar ga ketutupan php debugger
        let element = await
driver.findElement(By.className('sticky top-0 md:h-screen md:overflow-y-
auto px-5 py-8'));
        await driver.executeScript('arguments[0].scrollTop +=
700;', element);

        //pick color
        await driver.findElement(By.className('pcr-
button')).click();
        await driver.sleep(2000);

        const field = driver.findElement(By.className('pcr-
result'));
        field.click();
        field.sendKeys(Key.chord(Key.CONTROL, 'a')); //
select all text
        field.sendKeys(Key.BACK_SPACE); // or Key.DELETE to
delete all text

        await driver.findElement(By.className('pcr-
result')).sendKeys('#D51313');
        await driver.sleep(2000);
        await driver.findElement(By.className('pcr-
save')).click();
        await driver.sleep(2000);

```

```

        //validasi stroke
        await
driver.findElement(By.xpath("//div[contains(@class,'icons-
wrapper')]//*[contains(@style, 'stroke-width: 1.125')]"));
        //validasi warna icon
        await
driver.findElement(By.xpath("//div[contains(@class,'icons-
wrapper')]//*[contains(@style,'rgb(213, 19, 19)')]"));
        //validasi size icon
        const svgElement = await
driver.findElement(By.id("3rd-place")); // mengambil elemen SVG
        const widthAttribute = await
svgElement.getAttribute('width'); // mengambil nilai atribut width
        assert.equal(widthAttribute, '64'); // membandingkan
        nilai atribut dengan nilai yang diharapkan
    })
    })
})
})

```

Gambar 3.21 Skrip tes TC0119

Pada gambar 3.21 menunjukkan skrip tes TC0119 yang merupakan skenario pengujian mengenai sistem dapat menampilkan icon sesuai dengan format yang telah diatur oleh *member user*. Format tersebut berupa ukuran *icon*, *stroke icon*, dan juga warna *icon*. Pengujian dilakukan dengan memilih ukuran *icon* sebesar 64px, kemudian memilih *stroke icon* sebesar 3px, dan warna *icon* menjadi warna merah dengan *HEX code* #D51313 dan *RGB code* 213, 19, 19 kemudian dilakukan validasi untuk mengkonfirmasi apakah sistem dapat menerapkan format yang telah diatur oleh pengguna.

```

const {Builder, By, Key, until} = require('selenium-webdriver');
var expect = require('chai').expect;
require('chromedriver')
const loginModule = require('./Login/TC_0075');

describe('iconhub', function(){
  describe('Web', function(){
    describe('Member User', function(){
      before(async function(){
        await loginModule.login();
      })
      after(async function(){
        await driver.quit();
      })
      it('TC0121 - Sebagai member dapat memberi donasi', async
function() {

        //button donate
        var ele=
driver.wait(until.elementLocated(By.xpath('//*[@id="sidebar"]/div/div[3]
/div/div[4]/a[1]')));
        await ele.click();
        await driver.sleep(2000);

```

```

        await
driver.findElement(By.xpath('/html/body/div[1]/div/div[3]/div[5]/div/div
[2]/div[2]/a')).click();
        await driver.sleep(7000);

        // //validasi ada profil Bayu Winata
        var ele=
driver.wait(until.elementLocated(By.xpath('//div[contains(text(), "Bayu
Winata")]')));
        })
    })
})
})

```

Gambar 3.22 Skrip tes TC0121

Pada gambar 3.22 menunjukkan skrip tes TC0121 yang merupakan skenario pengujian mengenai member user dapat memberikan donasi secara sukarela atas *icon-icon* yang ada pada *website* Iconhub. Pengaksesan *website* Iconhub tidak dipungut biaya apapun alias gratis, begitu pula dengan mengunduh dan mengcopy *icon-icon* yang tersedia pada *website* Iconhub. Akan tetapi jika pengguna ingin mengapresiasi adanya *website* Iconhub ini maka pengguna dapat memberikan donasi kepada One Week Wonder sebagai penyedia *icon-icon* yang ada pada *website* Iconhub.

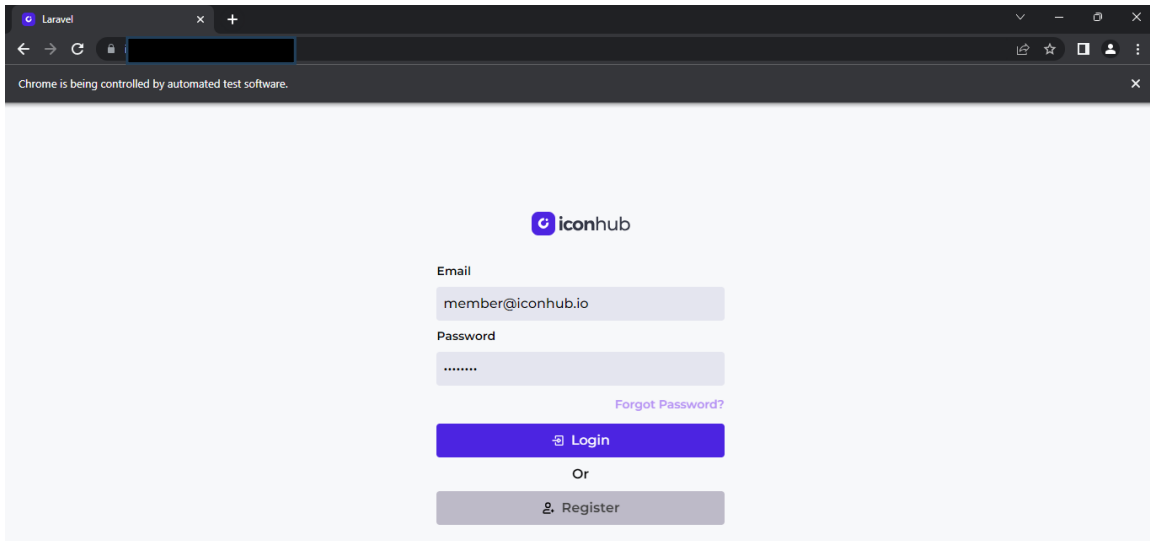
### 3.4.3 Eksekusi Skrip Tes

Setelah skrip tes dibuat, maka penguji perlu menjalankan skrip tersebut menggunakan perintah pada terminal seperti pada Gambar 3.23.

```
npm test .\[direktori]\[nama file].js
```

Gambar 3.23 Perintah menjalankan salah satu skrip tes

Setelah skrip tes dijalankan, maka akan muncul satu jendela *browser* secara otomatis yang menunjukkan bahwa pengujian otomatis sedang berlangsung. Adapun tampilan jendela *browser* ketika pengujian otomatis sedang berlangsung dapat dilihat pada Gambar 3.24.



Gambar 3.24 Tampilan halaman website Iconhub yang sedang diuji secara otomatis

Selanjutnya ketika pengujian otomatis telah selesai, maka akan terdapat hasil pengujian pada terminal, contohnya seperti pada Gambar 3.25.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\JAVAN\iconhub-selenium> npm test .\test\web-member_user\TC_0121.js

> selenium-iconhub@1.0.0 test
> mocha --no-timeouts .\test\web-member_user\TC_0121.js

iconhub
Web
Member User

DevTools listening on ws://127.0.0.1:61693/devtools/browser/3075f39b-576a-48a2-b428-ad3604f4abe7
✓ TC0121 - Sebagai member dapat memberi donasi (9185ms)

1 passing (27s)

```

Gambar 3.25 Hasil pengujian salah satu skrip tes

Pengujian yang dilakukan pada Gambar 3.25 adalah contoh dari *integration testing* karena pengujian tersebut menguji fungsi *button* Donate yang seharusnya dapat berintegrasi dengan halaman PayPal milik Bayu Winata.

Dalam satu *file* JavaScript dapat terdiri dari beberapa skenario pengujian. Skenario pengujian dibedakan menggunakan blok `'it'` ketika dituliskan pada skrip tes, jadi satu blok `'it'` berisi satu skenario pengujian, sehingga ketika pengujian menjalankan satu file skrip tes dapat sekaligus menguji lebih dari satu skenario pengujian, contoh dapat dilihat pada Gambar 3.26.



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\JAVAN\iconhub-selenium> npm test .\test\web-member_user\TC_0077_78_80.js

iconhub
Web
Member User

DevTools listening on ws://127.0.0.1:61823/devtools/browser/bf1a489f-9c9e-4c7b-b93c-1c52fea09077
✓ TC0077 - Sebagai member dapat memilih satu atau beberapa icon per icon style (11274ms)
✓ TC0078 - Sebagai member dapat download satu atau beberapa icon yang dipilih (6208ms)
Deselect All berhasil, tidak ada icon yang terpilih.
✓ TC0080 - Sebagai member dapat deselect semua icon yang dipilih (4112ms)

3 passing (33s)

```

Gambar 3.26 Hasil pengujian salah satu skrip tes dengan banyak kasus uji

Selenium juga dapat menjalankan *multiple* skrip tes dalam sekali perintah saja yaitu dengan menggunakan perintah yang hampir sama dengan Gambar 3.23 hanya saja tidak perlu menuliskan nama file.js, sehingga yang akan diuji yaitu kumpulan skrip tes yang terdapat pada salah satu folder. Adapun hasil eksekusi *multiple* skrip tes dengan Selenium WebDriver dapat dilihat pada Gambar 3.27.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Member User

DevTools listening on ws://127.0.0.1:64321/devtools/browser/48fd9c54-714f-4108-9118-18c48a17a3dc
✓ TC0096 - Sebagai member dapat mengatur warna dengan hex code (6409ms)
✓ TC0097 - Sebagai member dapat mengatur warna dengan rgb code (8534ms)

iconhub
Web
Member User

DevTools listening on ws://127.0.0.1:64343/devtools/browser/6fda1a95-8389-4a9a-b00f-0a9f79847aa0
✓ TC0112 - Sebagai member dapat menerapkan warna pada icon style (26600ms)

iconhub
Web
Member User
Web
Member User

DevTools listening on ws://127.0.0.1:64407/devtools/browser/52f54d8f-461a-49e8-b232-b53455f6898b
✓ TC0119 - Sebagai member dapat menampilkan icon sesuai dengan format (10891ms)

iconhub
Web
Member User

DevTools listening on ws://127.0.0.1:64440/devtools/browser/74445910-ac9b-41cd-a1bd-ac943ad629c2
✓ TC0121 - Sebagai member dapat memberi donasi (9168ms)

20 passing (6m)

```

Gambar 3.27 Hasil pengujian satu folder skrip tes

Pengujian *multiple test* yang hasilnya tertera pada Gambar 3.27 termasuk ke dalam jenis pengujian regresi karena pengujian tersebut menguji keseluruhan modul *role member user*. Percobaan pengujian regresi dengan alat pengujian otomatis pada *website* Iconhub telah berhasil dilakukan dengan status uji 100% *passed*. Pada Tabel 3.4 menunjukkan hasil pengujian regresi menggunakan Selenium WebDriver.

Tabel 3.4 Hasil pengujian regresi dengan Selenium WebDriver

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0076	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan <i>reset password</i>	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login yang berada di kanan atas halaman Iconhub</li> <li>3. Klik <i>text</i> "Forgot Password"</li> <li>4. <i>Input</i> email yang telah terdaftar</li> <li>5. Klik <i>button</i> Send Link Reset Password</li> <li>6. Klik <i>button</i> "Reset Password" pada email yang telah dikirim</li> <li>7. <i>Input password</i> baru dan konfirmasinya</li> <li>8. Klik <i>button</i> "Reset Password"</li> </ol>	<p>Tampil halaman untuk <i>request reset password</i></p> <p>Tampil <i>alert</i> "We have emailed your password reset link"</p> <p>Tampil halaman Reset Password</p> <p>Tampil halaman Login</p>	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0077	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat memilih satu atau banyak <i>icon</i> per <i>icon style</i>	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon Style, Size, Stroke</i> dan</li> </ol>	Dapat memilih <i>icon</i> yang diinginkan	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button</i> Select			
TC0078	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat mengunduh satu atau beberapa <i>icon</i>	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button</i> Select 8. Klik <i>button</i> Download ... <i>Icons</i> 9. Klik <i>button</i> ZIP atau SVG sesuai yang diinginkan	Terpilihnya <i>icon</i> Tampil pop up untuk memilih format file yang ingin diunduh Dapat mengunduh <i>icon-icon</i> yang dipilih sesuai format yang dipilih	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0079	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat <i>deselect</i> satu atau banyak <i>icon</i> per <i>icon</i> style	1. Akses halaman Iconhub 2. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password 4. Klik <i>button</i> Login 5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i>	Dapat memilih <i>icon</i> Dapat <i>deselect</i> pilihan <i>icon</i>	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			yang ingin dipilih 7. Klik <i>button</i> Select 8. <i>Hover mouse</i> ke arah <i>icon</i> yang telah dipilih sebelumnya 9. Klik <i>button</i> Deselect			
TC0080	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat <i>deselect</i> seluruh <i>icon</i> yang dipilih	1. Akses halaman <i>Iconhub</i> 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai 6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih 7. Klik <i>button</i> Select 8. Klik <i>button</i> Deselect All	Dapat <i>deselect</i> semua <i>icon</i> yang telah dipilih sebelumnya	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0081	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat <i>reset</i> filter ( <i>Icon</i> Style, Size, Stroke, dan Colors)	1. Akses halaman <i>Iconhub</i> 2. Klik <i>button</i> Login 3. <i>Input</i> email dan password 4. Klik <i>button</i> Login 5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai 6. Klik <i>Reset</i>	Tampil berbagai jenis <i>icon</i> dengan pengaturannya semula	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0082	<i>Select, Deselect, and Download Icons</i>	Sebagai <i>member user</i> dapat mengunduh semua <i>icon</i> per <i>Icon Style</i> dengan format <i>ZIP</i>	<ol style="list-style-type: none"> <li>1. Akses halaman <i>Iconhub</i></li> <li>2. Klik <i>button Login</i></li> <li>3. <i>Input</i> email dan password</li> <li>4. Klik <i>button Login</i></li> <li>5. Pilih <i>Icon Style, Size, Stroke dan Colors</i> yang sesuai</li> <li>6. Klik <i>button Download All Icons</i></li> </ol>	Dapat mengunduh semua <i>icon</i> yang ada pada <i>Icon Style</i> yang telah dipilih dengan format <i>ZIP</i>	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0083	<i>Search Icon</i>	Sebagai <i>member user</i> dapat mencari <i>icon</i> menggunakan filter <i>Pick From Our Icon Collection</i>	<ol style="list-style-type: none"> <li>1. Akses halaman <i>Iconhub</i></li> <li>2. Klik <i>button Login</i></li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button Login</i></li> <li>5. Pilih <i>Icon Style, Size, Stroke dan Colors</i> yang sesuai</li> <li>6. Pilih filter dari dropdown <i>Pick From Our Icon Collection</i></li> </ol>	Tampil berbagai jenis <i>icon</i> dengan filter yang telah diterapkan	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0084	<i>Search Icon</i>	Sebagai <i>member user</i> dapat menemukan <i>icon</i> yang diinginkan menggunakan <i>Search Bar</i>	<ol style="list-style-type: none"> <li>1. Akses halaman <i>Iconhub</i></li> <li>2. Klik <i>button Login</i></li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button Login</i></li> <li>5. Pilih <i>Icon Style, Size, Stroke dan Colors</i> yang sesuai</li> <li>6. <i>Inputkan</i> kata kunci <i>icon</i></li> </ol>	Dapat menampilkan <i>icon</i> sesuai dengan kata kunci yang dicari	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			yang ingin dicari			
TC0085	<i>Search Icon</i>	Sebagai <i>member user</i> tidak dapat menemukan <i>icon</i> yang tidak sesuai menggunakan Search Bar	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Inputkan</i> kata kunci yang tidak sesuai</li> </ol>	Tampil tulisan "Sorry, <i>Icon</i> Not Found" pada halaman <i>icon</i>	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0086	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat mengcopy <i>icon</i> dengan format yang sesuai	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Pilih <i>Icon</i> Style, Size, Stroke dan Colors yang sesuai</li> <li>6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih</li> <li>7. Klik <i>button</i> Copy SVG</li> </ol>	Dapat mengcopy <i>icon</i> dengan format yang sesuai	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0087	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan Logout	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login yang berada di kanan atas halaman Iconhub</li> <li>3. <i>Input</i> Email</li> </ol>	Tampil <i>pop up</i> Logout Dapat kembali ke halaman Login	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			<p>dan Password yang telah terdaftar</p> <p>4. Klik <i>button user profile</i> pada kanan atas</p> <p>5. Klik <i>pop up Logout</i></p>			
TC0088	<i>Select, Deselect, and Download Icons</i>	<p>Sebagai <i>member user</i> dapat memilih dan mengunduh <i>icon</i> yang sama pada <i>style</i> yang berbeda</p>	<ol style="list-style-type: none"> <li>1. Akses halaman <i>Iconhub</i></li> <li>2. Klik <i>button Login</i></li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button Login</i></li> <li>5. Pilih salah satu <i>Icon Style</i>, <i>Size</i>, <i>Stroke</i> dan <i>Colors</i></li> <li>6. <i>Hover mouse</i> ke arah <i>icon</i> yang ingin dipilih</li> <li>7. Klik <i>button Select</i></li> <li>8. Pilih <i>Icon Style</i> yang berbeda</li> <li>9. <i>Hover mouse</i> ke arah <i>icon</i> dengan nama yang sama dengan yang sebelumnya</li> <li>10. Klik <i>button Select</i></li> <li>11. Klik <i>button Download ... Icons</i></li> <li>12. Klik <i>button ZIP</i> atau <i>SVG</i> sesuai yang diinginkan</li> </ol>	<p>Tampil berbagai jenis <i>icon</i> dengan <i>style</i> yang berbeda dengan yang dipilih sebelumnya</p> <p>Dapat memilih <i>icon</i></p> <p>Tampil <i>pop up</i> untuk memilih format file yang ingin diunduh</p> <p>Dapat mengunduh <i>icon-icon</i> yang dipilih sesuai format yang dipilih walaupun dengan nama yang sama</p>	<p>Sesuai dengan hasil ekspektasi</p>	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0096	<i>Set Colors</i>	Sebagai <i>member user</i> dapat mengatur warna dengan hexcode	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>Color box</i></li> <li>6. Klik <i>button</i> HEXA</li> <li>7. <i>Input</i> hexcode yang diinginkan</li> </ol>	<p>Tampil pengaturan warna</p> <p>Tampil <i>field</i> untuk memasukkan hexcode</p> <p>Tampil warna yang sesuai dengan hexcode yang diinputkan</p>	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0097	<i>Set Colors</i>	Sebagai <i>member user</i> dapat mengatur warna dengan RGB	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>Color box</i></li> <li>6. Klik <i>button</i> RGBA</li> <li>7. <i>Input</i> kode rgb yang diinginkan</li> </ol>	<p>Tampil pengaturan warna</p> <p>Tampil <i>field</i> untuk memasukkan kode warna RGB</p> <p>Tampil warna yang sesuai dengan kode RGB yang diinputkan</p>	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0112	<i>Set Colors</i>	Sebagai <i>member user</i> dapat menyimpan warna <i>primary</i> , <i>secondary</i> , dan <i>tertiary</i> untuk diterapkan ke semua style	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>Icon Style</i> "Multi Color"</li> <li>6. Klik <i>Color Box Primary</i></li> <li>7. Pilih warna yang diinginkan</li> <li>8. Klik <i>button</i> Save</li> <li>9. Klik <i>Color</i></li> </ol>	<p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>primary</i></p> <p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>secondary</i></p> <p>Tampil pengaturan warna</p> <p>Dapat menyimpan warna yang dipilih untuk warna <i>tertiary</i></p>	Sesuai dengan hasil ekspektasi	<i>Passed</i>



<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
			Box Secondary 10. Pilih warna yang diinginkan 11. Klik <i>button</i> Save 12. Klik Color Box Tertiary 13. Pilih warna yang diinginkan 14. Klik <i>button</i> Save			
TC0116	<i>Basic Functions</i>	Sebagai <i>member user</i> tidak dapat melakukan registrasi dengan email yang telah terdaftar	1. Akses halaman Iconhub 2. Klik <i>button</i> login yang berada di kanan atas halaman Iconhub 3. Klik <i>button</i> Register 4. <i>Input</i> Email yang sudah terdaftar 5. Klik <i>button</i> Register	Tampil <i>alert</i> "The email has already been taken."	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0118	<i>Login, Logout, and Reset Password</i>	Sebagai <i>member user</i> dapat melakukan Login melalui halaman Register	1. Akses halaman Iconhub 2. Klik <i>button</i> login yang berada di kanan atas halaman Iconhub 3. Klik <i>button</i> register 4. Klik <i>button</i> Login 3. <i>Input</i> Email dan Password yang telah terdaftar 5. Klik <i>button</i> Login	Tampil halaman Login Tampil halaman Register Tampil halaman Login Tampil <i>pop up</i> berhasil login dan <i>home page</i> sebagai <i>Member user</i>	Sesuai dengan hasil ekspektasi	<i>Passed</i>

<i>Case Code</i>	<i>Category</i>	<i>Description</i>	<i>Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TC0119	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat menampilkan <i>icon</i> sesuai dengan <i>icon style</i> dan format ( <i>size</i> , <i>stroke</i> , dan <i>colors</i> ) yang telah diatur	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>icon style</i> yang diinginkan</li> <li>6. Klik <i>Size</i> yang diinginkan</li> <li>7. Klik <i>Stroke</i> yang diinginkan</li> <li>8. Klik <i>color box</i> dan pilih warna yang diinginkan</li> </ol>	Tampil jenis <i>icon style</i> yang dipilih Tampil <i>icon</i> sesuai dengan <i>size</i> yang dipilih Tampil <i>icon</i> sesuai dengan <i>stroke</i> yang dipilih Tampil <i>icon</i> dengan warna yang sesuai	Sesuai dengan hasil ekspektasi	<i>Passed</i>
TC0121	<i>Basic Functions</i>	Sebagai <i>member user</i> dapat memberi donasi	<ol style="list-style-type: none"> <li>1. Akses halaman Iconhub</li> <li>2. Klik <i>button</i> Login</li> <li>3. <i>Input</i> Email dan Password</li> <li>4. Klik <i>button</i> Login</li> <li>5. Klik <i>button</i> Donate</li> <li>6. Klik <i>button</i> Donate via Paypal</li> </ol>	Tampil <i>pop up</i> "Support Our Work" Diarahkan ke <i>tab</i> baru halaman Paypal milik Bayu Winata dengan <i>username</i> @bayuhendra	Sesuai dengan hasil ekspektasi	<i>Passed</i>

#### 3.4.4 Integrasi Allure Framework Report dengan Selenium WebDriver

Berdasarkan hasil dari penelitian-penelitian sebelumnya (Gojare et al., 2015; Ramya P et al., 2017), telah dikatakan bahwa Selenium WebDriver tidak memiliki fungsi untuk menghasilkan laporan atau *report* hasil pengujian yang telah dilakukan. Laporan yang dihasilkan oleh Selenium WebDriver hanya tertera pada terminalnya dan kurang informatif, sehingga akan lebih baik jika Selenium WebDriver dapat diintegrasikan dengan *add on* atau *third party tools* penghasil laporan, seperti Allure Framework Report.

Allure Framework Report adalah sebuah kerangka kerja yang digunakan untuk menghasilkan laporan pegujian yang informatif dan visual secara otomatis. Laporan yang

dihasilkan oleh Allure mencakup berbagai informasi penting seperti hasil pengujian, statistik, kesalahan, skenario yang berhasil, dan visualisasi grafis yang dapat membantu dalam meningkatkan transparansi pengujian, memfasilitasi analisis hasil pengujian, dan mendukung pengambilan keputusan yang lebih baik dalam proses pengembangan perangkat lunak.

### 3.4.5 Instalasi dan Eksekusi Skrip Tes dengan Allure Framework Report

Sebelum dapat menghasilkan laporan hasil pengujian, harus dilakukan instalasi untuk mempersiapkan Allure. Langkah pertama adalah penginstalan *dependencies* untuk allure-report dengan menjalankan perintah pada terminal VSCode seperti pada Gambar 3.28.

```
npm install allure-commandline
npm install mocha-allure-reporter
```

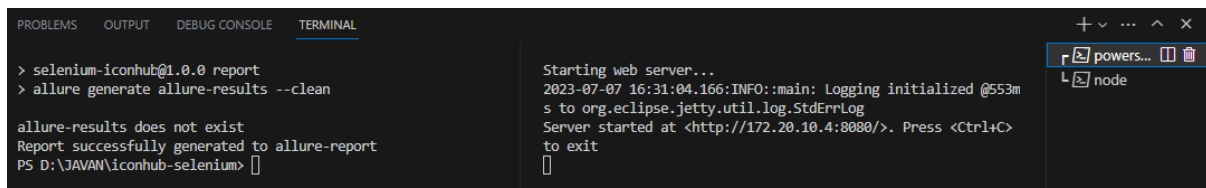
Gambar 3.28 Perintah instalasi *dependencies* allure-report

Setelah dilakukan instalasi *dependencies* allure-report pastikan skrip pada *file* package.json seperti Gambar 3.29.

```
"scripts": {
  "test": "mocha -r dotenv/config --no-timeouts --reporter mocha-allure-reporter",
  "start": "allure open allure-report --port 8080",
  "report": "allure generate allure-results --clean"
},
```

Gambar 3.29 Penulisan skrip pada *file* package.json

Kemudian setelah memastikan skrip pada *file* package.json sudah sesuai, maka jalankan `npm run report` pada terminal VSCode untuk menghasilkan folder allure-report. Selanjutnya jalankan `npm run start` pada terminal untuk meluncurkan allure-report pada *browser* dengan *port* 8080 secara otomatis, terminal ini jangan ditutup ataupun diberhentikan agar *server* allure-report tetap berjalan. Adapun tampilan terminal setelah menjalankan dua perintah diatas menjadi seperti pada Gambar 3.30 sedangkan tampilan halaman Allure Report yang sudah diluncurkan pertama kali seperti pada Gambar 3.31.

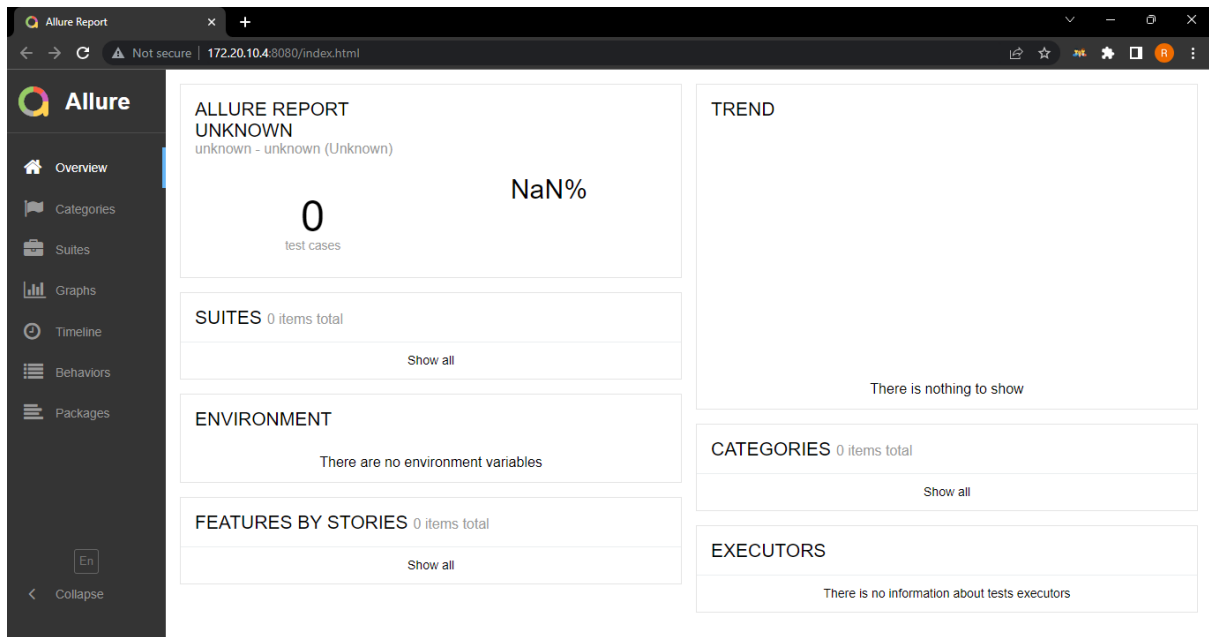


```
> selenium-iconhub@1.0.0 report
> allure generate allure-results --clean

allure-results does not exist
Report successfully generated to allure-report
PS D:\JAVAN\iconhub-selenium>

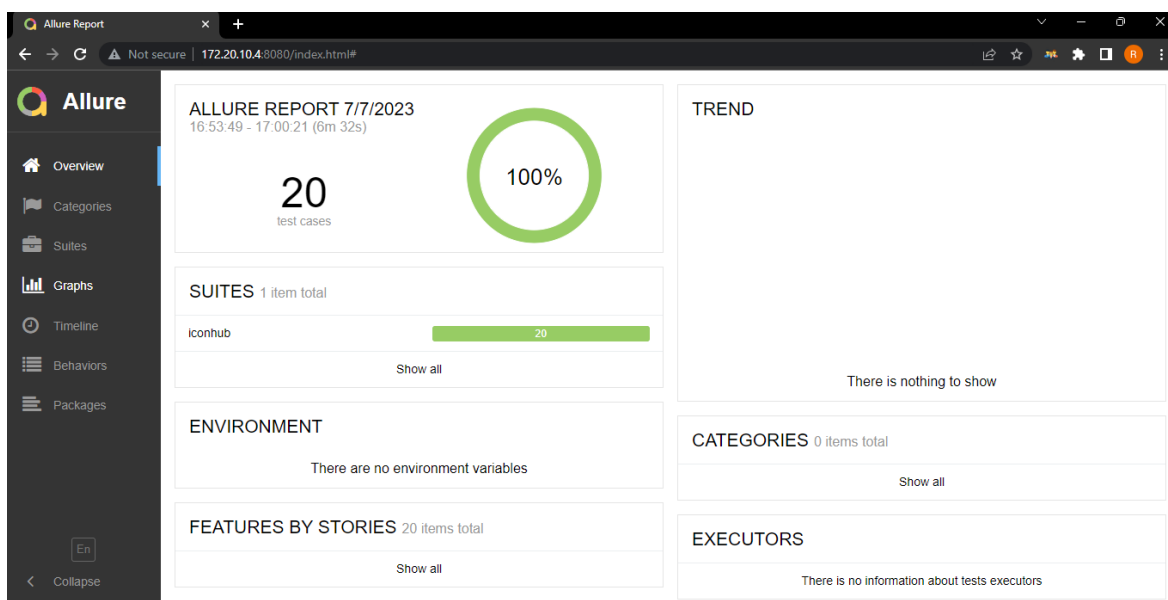
Starting web server...
2023-07-07 16:31:04.166:INFO:main: Logging initialized @553ms
to org.eclipse.jetty.util.log.StdErrLog
Server started at <http://172.20.10.4:8080/>. Press <Ctrl+C>
to exit
```

Gambar 3.30 Tampilan terminal VSCode



Gambar 3.31 Tampilan awal halaman Allure Report pada *browser*

Setelah itu jalankan skrip tes Selenium yang diinginkan pada terminal lain, bukan terminal yang meluncurkan *browser* dengan *port* 8080. Skrip tes Selenium yang dijalankan akan menghasilkan folder *allure-results* secara otomatis. Selanjutnya setelah pengujian dilaksanakan, maka jalankan perintah `npm run report` kembali dan *refresh* halaman *browser* dengan *port* 8080. Setelah itu maka akan muncul hasil pengujian yang telah dilakukan pada halaman *browser* Allure Report seperti Gambar 3.32 dan Gambar 3.33.

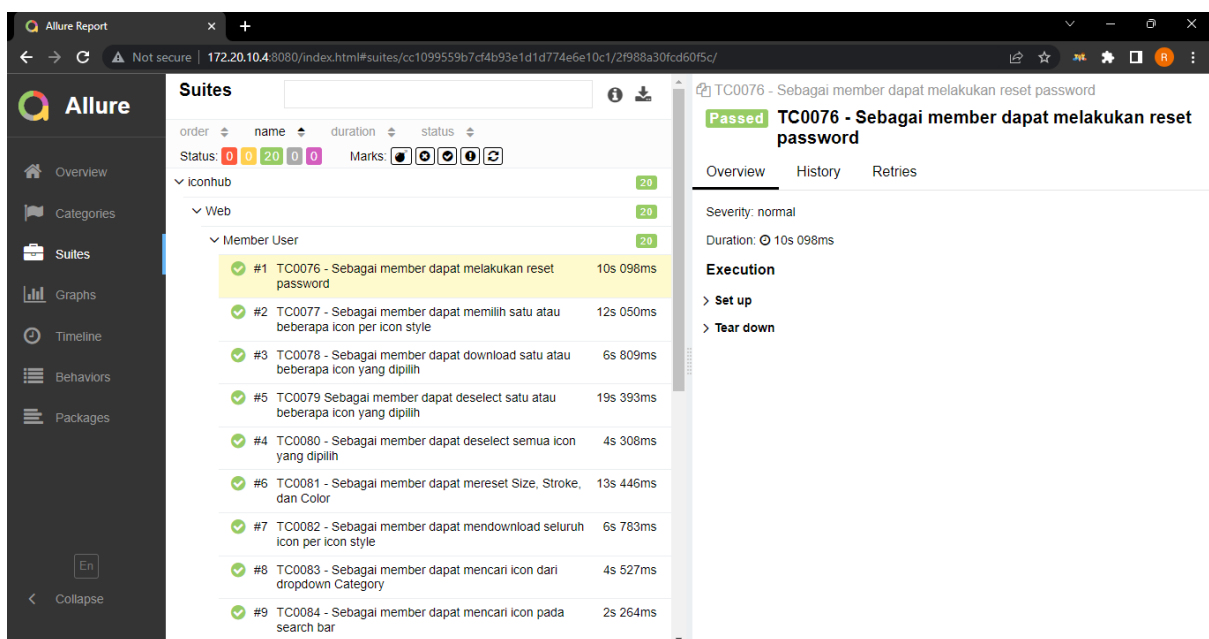


Gambar 3.32 Tampilan halaman Overview hasil pengujian regresi pada Allure Report

Pada Gambar 3.32 yaitu tampilan halaman Overview hasil pengujian regresi pada Allure Report terdapat beberapa bagian, yaitu:

1. Visualisasi grafik status hasil pengujian.
2. Suites digunakan untuk mengelompokkan skenario pengujian menjadi kelompok yang lebih terstruktur berdasarkan kriteria tertentu.
3. Environment mengacu pada informasi tentang konfigurasi lingkungan pengujian.
4. Features by Stories adalah cara untuk mengorganisir pengujian berdasarkan fitur atau cerita pengguna.
5. Trend adalah bagian dari laporan Allure yang menunjukkan perkembangan hasil pengujian dari waktu ke waktu.
6. Categories adalah cara untuk mengelompokkan pengujian berdasarkan kategori tertentu yang ditentukan oleh pengguna.
7. Executors adalah bagian dari Allure Report yang menunjukkan detail tentang pelaksanaan atau eksekusi pengujian.

Semua komponen yang telah disebutkan diatas dapat berfungsi jika dikonfigurasi dengan *plugin* Allure.



Gambar 3.33 Tampilan halaman Suites hasil pengujian regresi pada Allure Report

Sedangkan pada Gambar 3.33 yaitu tampilan halaman Suites hasil pengujian regresi pada Allure Report yang di dalamnya juga terdapat beberapa bagian, yaitu pengelompokkan skenario pengujian secara informatif seperti judul skenario pengujian beserta waktu eksekusi, statistik status hasil pengujian yang dibedakan menjadi lima status, yaitu berhasil, gagal, rusak, terlewat, dan yang tidak diketahui serta menyajikan informasi terkait secara terpisah disebelah kanan halaman Suite Allure Report.

### **3.5 Penutupan Proyek**

Proyek pengembangan *website* Iconhub resmi ditutup pada tanggal 12 Mei 2023 bertepatan dengan masa berakhirnya kontrak OWW (One Week Wonder) dengan PT. Javan Cipta Solusi. PT. Javan Cipta Solusi telah menyelesaikan tugas beserta tanggung jawab sebagai penyedia *platform* sekaligus pengembang *website* Iconhub dengan baik. Selama proyek berlangsung, PT. Javan Cipta Solusi telah melakukan *User Acceptance Testing* (UAT) sebanyak 3 kali dan selalu dilakukan *follow up* atas *feedback* yang telah diberikan. Berakhirnya proyek *website* Iconhub tidak berarti selesainya pengembangan dari *website* tersebut, karena sistem atau aplikasi yang baik tidak akan pernah berhenti dikembangkan.

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Relevansi Akademik

Berdasarkan teori yang telah ditinjau sebelumnya, penulis menemukan kesenjangan antara teori dengan pelaksanaan di lapangan ketika magang. Terdapat beberapa hal yang akan dibahas pada sub-bab ini, yaitu terkait Scrum, pengujian regresi, dan pengujian otomatis menggunakan Selenium WebDriver.

##### 4.1.1 *Scrum*

PT. Javan Cipta Solusi merupakan perusahaan yang sangat menerapkan metode Scrum dengan benar, seperti yang disebutkan dalam buku karya Kenett & Baker (2010) bahwa ada beberapa *key practices* seperti berfokus pada *sprint*, pertemuan, *review*, dan jadwal proyek Tahapan-tahapan dalam metode Scrum yang disebutkan pada teori dijalankan dengan baik di PT. Javan Cipta Solusi, hanya saja selama penulis magang tidak mendapatkan tahapan Sprint Retrospective. Kemudian pada buku yang ditulis oleh Woodward et al., (2010), disebutkan ada tiga peranan pada metode *Scrum*, yaitu *product owner*, *scrum master*, dan tim. Pada kenyataannya *scrum master* pada PT. Javan Cipta Solusi dirangkap oleh satu peran yang disebut Tribe Lead, sehingga cakupan tanggung jawabnya lebih besar daripada seorang *scrum master*.

##### 4.1.2 Pengujian Regresi

Ppengujian regresi adalah jenis pengujian perangkat lunak yang memiliki tujuan untuk memastikan bahwa perubahan yang dibuat pada saat pengembangan perangkat lunak tidak mempengaruhi fitur perangkat lunak yang seharusnya tidak berubah (Wong et al., 1997), sehingga dapat disimpulkan bahwa pengujian regresi bersifat menyeluruh yang artinya pengujian harus dilakukan dari awal hingga akhir. Pada kenyataannya, di lapangan masih banyak yang salah mengartikan kegiatan ‘meregresi’, sehingga banyak yang melakukan pengujian regresi hanya pada bagian tertentu yang dirasa berubah. Hal itu dikarenakan manusia memiliki rasa malas sehingga hanya meregresi bagian tertentu, maka dari itu diperlukan pengujian otomatis yang akan sangat membantu proses pengujian regresi secara menyeluruh tanpa memerlukan energi dan waktu yang banyak. Jika hanya melakukan pengujian pada beberapa fitur yang saling berintegrasi dinamakan *integration testing* seperti disebutkan dalam penelitian oleh Anand & Uddin (2019) yang menyebutkan bahwa *integration testing* adalah pengujian yang melibatkan pengujian interaksi antara unit-unit sistem.

### 4.1.3 Pengujian Otomatis menggunakan Selenium WebDriver

Keuntungan menggunakan Selenium WebDriver disebutkan salah satunya yaitu bahwa Selenium WebDriver tidak masalah dengan halaman web yang bersifat dinamis (Vila et al., 2017). Pada kenyataannya, memang Selenium WebDriver dapat mengatasi pengujian pada halaman web yang bersifat dinamis, tetapi penguji harus berhati-hati pada penulisan skrip kodenya dan harus mengetahui fungsi Selenium WebDriver yang tepat untuk digunakan pada pengujian otomatis terhadap halaman web yang dinamis.

Selenium WebDriver bekerja dengan mengambil elemen-elemen yang ada pada web *browser*, sehingga penguji harus memastikan elemen yang seharusnya berfungsi tidak tertutupi oleh elemen lain. Jika suatu elemen tertutupi oleh elemen lain, maka *driver* Selenium tidak akan dapat menemukan elemen yang seharusnya dicari, sehingga pengujian tidak dapat terselesaikan atau bahkan menjadi gagal karena yang menerima perintah bukan elemen yang seharusnya.

Secara teori, dikatakan bahwa pengujian otomatis dapat menghemat waktu (Hanna et al., 2018). Pada kenyataannya persiapan untuk pengujian otomatis menggunakan Selenium WebDriver memakan banyak waktu. Penulisan skrip kode pengujian memakan banyak waktu apalagi jika seorang penguji belum mahir dalam pemrograman, tetapi ketika skrip tes sudah selesai dibuat maka pengujian otomatis terbukti dapat mempercepat proses pengujian perangkat lunak.

Sebelum pembuatan skrip tes pengujian otomatis menggunakan Selenium WebDriver seharusnya *test case* atau skenario pengujian dilakukan verifikasi terlebih dahulu oleh sistem analis atau *product owner*, sehingga skenario pengujian yang berpengaruh pada skrip tes yang akan dibuat sudah terjamin sesuai dengan *acceptance criteria* perangkat lunak. Pada kenyataan yang terjadi, tidak ada tahapan verifikasi skenario pengujian tersebut.

## 4.2 Pembelajaran Magang

Magang dapat menjadi salah satu kegiatan yang bermanfaat bagi perkembangan diri seseorang, tidak hanya perkembangan diri tetapi juga perkembangan karir. Magang memberikan wawasan praktis dan pengalaman kerja yang berharga untuk dapat mempersiapkan diri dalam memasuki dunia kerja secara penuh.

Sebelum penulis magang pada PT. Javan Cipta Solusi, penulis telah mengikuti program Magang Bersertifikat Kampus Merdeka di suatu perusahaan yang bergerak di bidang ERP (*Enterprise Resource Planning*) yang terletak di Jakarta Pusat selama kurang lebih 4,5 bulan.



Sayangnya perusahaan tersebut mengharuskan pemegang dan karyawan yang bekerja maupun pernah bekerja disana untuk menandatangani perjanjian kerahasiaan karyawan, sehingga penulis tidak diperkenankan untuk menuliskannya ke dalam laporan akhir. Pada akhirnya penulis harus mencari tempat magang lagi untuk menggenapi wajib magang 6 bulan dan mencari tempat magang yang mendukung penulis untuk menuliskan kegiatannya selama magang ke dalam laporan akhir. Penulis sangat bersyukur dapat diterima di PT. Javan Cipta Solusi dan mendapat pekerjaan yang lebih menantang daripada pekerjaan sebelumnya.

Setelah menyelesaikan magang pada PT. Javan Cipta Solusi sebagai seorang Quality Assurance, penulis mendapatkan banyak manfaat baik dari segi akademik dan non akademik. Di PT. Javan Cipta Solusi, penulis belajar hal baru yaitu tentang menjalankan pengujian otomatis dengan Selenium WebDriver berbahasa JavaScript. Pada saat duduk dibangku perkuliahan, penulis hanya pernah berkenalan dengan salah satu alat penguji otomatis, yaitu Selenium IDE. Selama magang, penulis banyak belajar tentang bagaimana menulis skrip tes dan menjalankannya dengan Selenium WebDriver. Di awal perkenalan dengan Selenium WebDriver, Lead QA membagikan *link* Github yang berisi skrip tes Selenium berbahasa JavaScript yang sudah pernah ditulis sebelumnya, sehingga dari skrip tes Selenium tersebut penulis belajar memahami struktur penulisan skrip tes Selenium berbahasa JavaScript. Selenium WebDriver memiliki banyak pustaka yang berisi fungsi, kelas, dan objek yang dapat digunakan untuk menuliskan skrip tes. Penulis juga belajar menggunakan *tools-tools* baru, yaitu alat manajemen proyek seperti Taiga dan ActiveCollab, alat manajemen tes seperti QATouch, dan alat manajemen repositori seperti GitLab. Penulis juga berkesempatan untuk berkenalan dengan alat penguji otomatis lainnya seperti JMeter dan Postman, walaupun sayangnya penulis belum dapat menggunakannya dalam proyek yang penulis berpartisipasi di dalamnya.

Pada saat mempelajari tentang Selenium WebDriver penulis menemukan kendala, hambatan serta tantangan tersendiri yang belum pernah penulis temukan sebelumnya. Beberapa diantaranya seperti ketika mencari elemen untuk dapat diklik atau ketika element tidak dapat diklik oleh pengujian otomatis dikarenakan elemen tersebut tertutup elemen lainnya. Hal tersebut sangat berpengaruh terhadap proses pengujian otomatis dengan Selenium WebDriver dan terkadang hal-hal tersebut yang menjadi kendala bagi penulis menyelesaikan skrip tes dengan cepat. Untuk dapat mengatasi hal-hal tersebut penulis dapat bertanya kepada Mentor dan/atau rekan-rekan QA lainnya secara langsung ataupun melalui *chat*. Tidak hanya dengan bertanya kepada Mentor atau rekan-rekan QA, penulis juga memanfaatkan ChatGPT untuk

mencari solusi dari kendala, hambatan, dan tantangan yang penulis jumpai. Dengan ChatGPT, penulis mempelajari banyak hal baru mengenai Selenium WebDriver terutama mengenai fungsi-fungsi yang dapat digunakan untuk penulisan skrip tes.

Penulis juga pernah mengalami hal yang kurang mengenakkan yaitu pada waktu penulis harus membuat dokumen *user guide* untuk *website* Iconhub tetapi *product owner* baru memberi tahu H-1 sebelum dilakukan *sprint review*, sehingga dokumen *user guide* masih banyak kekurangannya dan menjadi tugas tambahan pada *sprint* selanjutnya. Hal tersebut terjadi dikarenakan penulis baru bergabung pada proyek *website* Iconhub tetapi tidak diberi tahu pekerjaan yang sudah disepakati pada *sprint planning* yang telah dilaksanakan sebelum penulis bergabung ke dalam proyek. Walaupun begitu, penulis dapat menyelesaikan dokumen *user guide* dengan tuntas pada *sprint* berikutnya.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pengujian regresi sebuah sistem dapat memanfaatkan alat pengujian otomatis Selenium WebDriver dibantu dengan Allure Framework Report untuk menghasilkan laporan hasil pengujian yang informatif. Pada laporan akhir ini telah berhasil dilakukan percobaan pengujian otomatis menggunakan Selenium WebDriver pada sebuah *website* penyedia layanan *icon* bernama Iconhub dengan presentasi lulus uji 100%. Percobaan pengujian tersebut berhasil mengeksekusi 20 skrip tes dengan waktu total kurang lebih enam menit. Dengan begitu maka terbukti penggunaan pengujian otomatis Selenium WebDriver dapat mempercepat proses pengujian perangkat lunak dan membuat pengujian tersebut menjadi lebih efektif dan efisien.

Akan tetapi kecepatan Selenium WebDriver dalam melakukan pengujian regresi dapat dipengaruhi oleh kecepatan internet dan kemampuan web *browser* dalam melakukan proses pengujian. Pengujian otomatis menggunakan Selenium WebDriver juga perlu disesuaikan kembali saat terjadi pengembangan pada sistem, karena bisa jadi elemen-elemen yang sudah dituliskan pada skrip tes berubah, sehingga jika pengujian dijalankan kembali maka hasilnya tidak akan lulus uji 100%.

#### **5.2 Saran**

Terdapat saran yaitu untuk melakukan perbandingan kecepatan alat pengujian otomatis lainnya dalam melakukan pengujian regresi. Dengan melakukan perbandingan alat pengujian otomatis lainnya, maka akan mengetahui alat pengujian mana yang paling efisien untuk pengujian regresi sebuah aplikasi atau sistem. Selain itu, terdapat saran untuk melakukan integrasi *reporting tools* pihak ketiga selain Allure Framework Report, sehingga dapat mengetahui *reporting tools* yang tepat untuk Selenium WebDriver.

Kemudian saran untuk pengujian yang dilakukan oleh QA sebaiknya *test case* atau skenario pengujian untuk dapat diverifikasi terlebih dahulu oleh sistem analis atau product owner sebelum dilakukan pengujian regresi, sehingga *test case* atau skenario pengujian yang dibuat terverifikasi sesuai dengan *acceptance criteria* yang telah dibuat oleh sistem analis.

### DAFTAR PUSTAKA

- Amalia, A., & Cahyono, A. B. (2022). *Analisis Pemanfaatan Playwright untuk Pengujian Otomatis Aplikasi Berbasis Web (Studi Kasus: Sistem Manajemen Jaringan)*.
- Anand, A., & Uddin, A. (2019). Importance of Software Testing in the Process of Software Development Cloud Security View project Cloud Computing in Load balancing View project Importance of Software Testing in the Process of Software Development. In *IJSRD-International Journal for Scientific Research & Development/* (Vol. 6). www.ijsrd.com
- Bruns A, Kornstadt A, & Wichmann D. (2009). *Web Application Tests with Selenium*. IEEE Software. <https://doi.org/10.1109/ms.2009.144>
- García, B., Gallego, M., Gortázar, F., & Munoz-Organero, M. (2020). A survey of the selenium ecosystem. *Electronics (Switzerland)*, 9(7), 1–29. <https://doi.org/10.3390/electronics9071067>
- Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341–346. <https://doi.org/10.1016/j.procs.2015.04.038>
- Hanna, M., Aboutabl, A. E., & Mostafa, M.-S. M. (2018). Automated Software Testing Framework for Web Applications. In *International Journal of Applied Engineering Research* (Vol. 13, Issue 11). <http://www.ripublication.com>
- Kenett, R. S., & Baker, E. R. (2010). *Process Improvement and CMMI® for Systems and Software*. Auerbach Publications. <https://doi.org/10.1201/9781420060515>
- Kosasih, Y., & Cahyono, A. B. (2021). *Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale (Studi Kasus TPOS PT. JAVASIGNA INTERMEDIA)*.
- Permana, P. A. G. (2015). Scrum Method Implementation in a Software Development Project Management. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 6, Issue 9). www.ijacsa.thesai.org
- Prabowo D, & Putro H. (2019). Pengujian Regresi Otomatis Menggunakan Selenium IDE. *Prosiding SENDIPA*, VII, 49–54. [https://www.academia.edu/90069188/Pengujian\\_Regresi\\_Otomatis\\_Menggunakan\\_Selenium\\_IDE](https://www.academia.edu/90069188/Pengujian_Regresi_Otomatis_Menggunakan_Selenium_IDE)
- Ramya P, Sindhura V, & Sagar P. (2017). *Testing Using Selenium Web Driver*.

- Vila, E., Novakova, G., & Todorova, D. (2017). Automation testing framework for web applications with selenium WebDriver: Opportunities and threats. *ACM International Conference Proceeding Series, Part F131200*, 144–150. <https://doi.org/10.1145/3133264.3133300>
- Wijaya, A., Fasa, H., & Sani, S. Y. (2020). Sistem Manajemen Anti-Penyuapan ISO 37001:2016 dan Pencegahan Praktik Korupsi di Sektor Pelayanan Publik. *INTEGRITAS: Jurnal Antikorupsi*, 6(2), 187–208. <https://doi.org/10.32697/integritas.v6i2.684>
- Wong, W. E., Horgan, J. R., London, S., & Agrawal, H. (1997). *A Study of Effective Regression Testing in Practice\**.
- Woodward, E., Surdek, S., & Ganis, M. (2010). *A Practical Guide to Distributed Scrum* (1st ed.). IBM Press.

## **LAMPIRAN**