

**PEMANFAATAN CYPRESS UNTUK  
PENGUJIAN END-TO-END (STUDI KASUS:  
PENGEMBANGAN APLIKASI INDICAR)**



Disusun Oleh:

N a m a : Bima Prakoso

NIM : 19523151

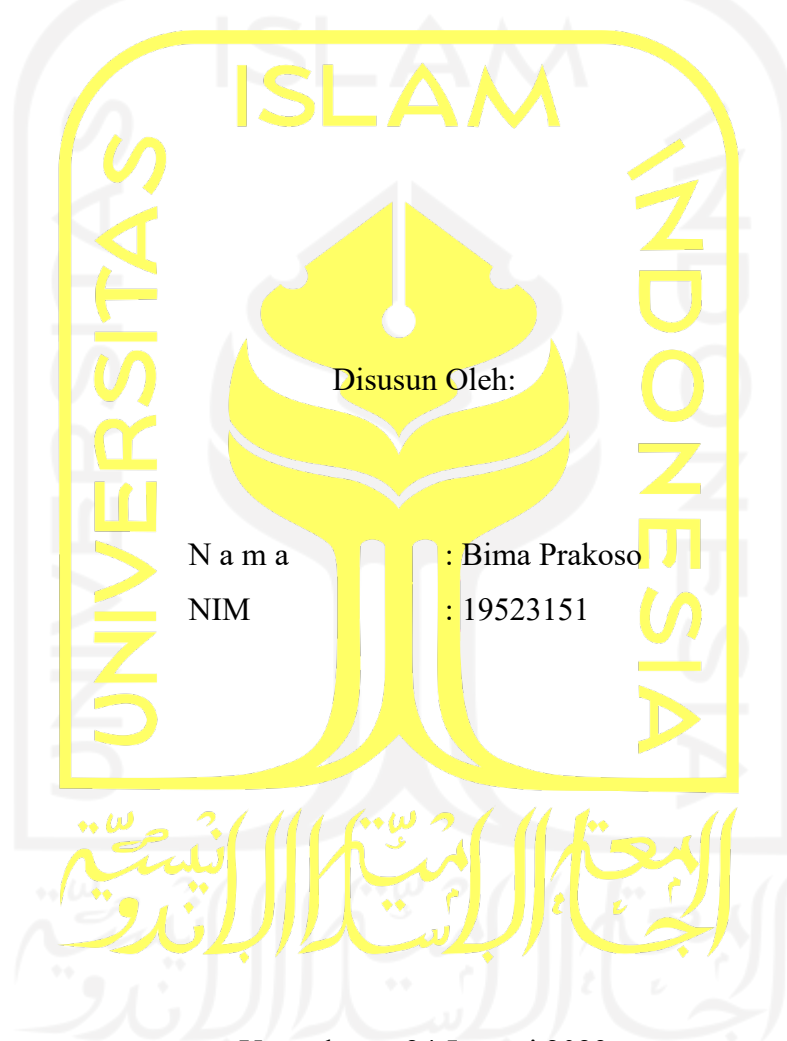
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2023**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

**PEMANFAATAN CYPRESS UNTUK**  
**PENGUJIAN END-TO-END (STUDI KASUS:**  
**PENGEMBANGAN APLIKASI INDICAR)**

**TUGAS AKHIR JALUR MAGANG**



Yogyakarta, 24 Januari 2023

Pembimbing,

( Ari Sujarwo, S.Kom, M.IT. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**PEMANFAATAN CYPRESS UNTUK**  
**PENGUJIAN END-TO-END (STUDI KASUS:**  
**PENGEMBANGAN APLIKASI INDICAR)**

**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, 24 Januari 2023

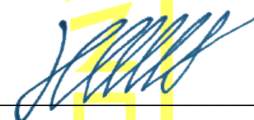
Tim Penguji

Ari Sujarwo, S.Kom, M.IT.



**Anggota 1**

Hanson Prihantoro Putro, S.T., M.T.



**Anggota 2**

Rahadian Kurniawan, S.KOM., 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 : Bima Prakoso  
NIM : 19523151

Tugas akhir dengan judul:

**PEMANFAATAN CYPRESS UNTUK  
PENGUJIAN END-TO-END (STUDI KASUS:  
PENGEMBANGAN APLIKASI INDICAR)**

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 apa pun.

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

Yogyakarta, 24 Januari 2023



( Bima Prakoso )

## HALAMAN PERSEMBAHAN

Tugas akhir ini penulis persembahkan untuk kedua orangtua yang telah mendukung, membantu baik secara mental dan doa, serta menjadi tempat berkeluh kesah selama masa perkuliahan. Tidak hanya itu, tugas akhir ini penulis persembahkan untuk sahabat-sahabat terbaikku yang sering memberikan masukan, dukungan, serta motivasi selama membuat tugas akhir ini. Kalian sudah menjadi teman terbaik untuk penulis selama menempuh pendidikan sarjana.



## HALAMAN MOTO

“Orang yang hebat adalah orang yang memiliki kemampuan menyembunyikan kesusahan, sehingga orang lain mengira bahwa ia selalu senang.”

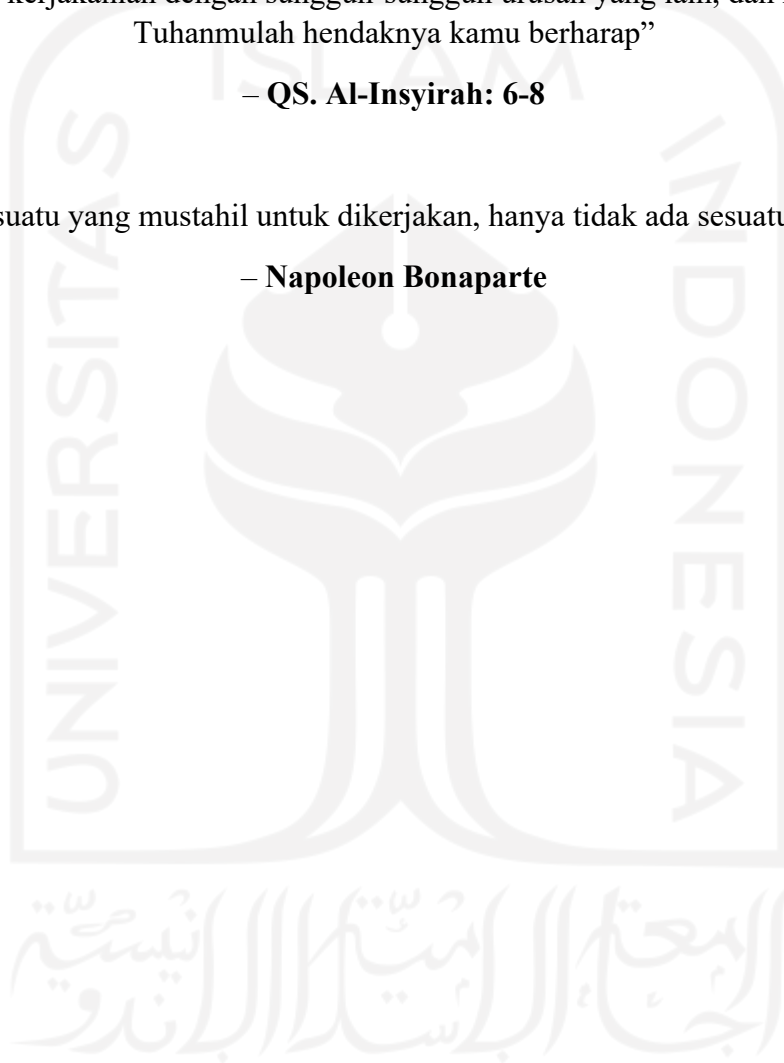
– **Imam Syafi’i**

“Sesungguhnya sesudah kesulitan itu ada kemudahan, maka apabila kamu telah selesai dari suatu urusan, kerjakanlah dengan sungguh-sungguh urusan yang lain, dan hanya kepada Tuhanmulah hendaknya kamu berharap”

– **QS. Al-Insyirah: 6-8**

“Tidak ada sesuatu yang mustahil untuk dikerjakan, hanya tidak ada sesuatu yang mudah”

– **Napoleon Bonaparte**



## KATA PENGANTAR

Dengan memanjatkan puja dan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul **“Pemanfaatan Cypress untuk Pengujian End-to-end (Studi Kasus: Pengembangan Aplikasi IndiCar)”**, sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Informatika Universitas Islam Indonesia.

Penulis menyadari bahwa skripsi ini tidak mungkin terselesaikan tanpa adanya dukungan, bantuan, bimbingan, dan nasehat dari berbagai pihak selama penyusunan skripsi ini. Pada kesempatan ini, penulis menyampaikan terima kasih kepada:

1. Allah SWT yang telah memberikan kesehatan, kemudahan, serta kekuatan.
2. Keluarga khususnya kedua orangtua yang selalu memberikan do'a, dukungan, serta kepercayaan penuh kepada Penulis.
3. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku ketua prodi Informatika Universitas Islam Indonesia.
4. Bapak Ari Sujarwo, S.Kom, M.IT. selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk membimbing dan mengarahkan selama proses magang hingga dalam penyelesaian Tugas Akhir.
5. Pimpinan, supervisor dan tim PT Telkom Indonesia yang telah memberikan ilmu, kepercayaan serta bimbingan selama proses magang berlangsung.
6. Dosen Program Studi Informatika Universitas Islam Indonesia yang telah memberikan bekal ilmu kepada penulis selama perkuliahan.
7. Teman seperjuangan kuliah dan magang yang saling membantu, mendukung, dan meluangkan waktu untuk mendengarkan keluh kesah selama perkuliahan.

Atas bantuan berbagai pihak, penulis dapat menyelesaikan laporan tugas akhir. Akan tetapi, laporan ini masih jauh dari kata sempurna sehingga membutuhkan saran untuk menyempurnakan laporan ini. Akhir kata, semoga laporan ini dapat bermanfaat bagi pembaca. Terima kasih.

Yogyakarta, 24 Januari 2023



( Bima Prakoso )

## SARI

PT Telkom Indonesia (Persero) Tbk (Telkom) adalah perusahaan yang bergerak di bidang jasa layanan teknologi informasi dan komunikasi (TIK) dan jaringan telekomunikasi di Indonesia yang dimiliki oleh negara atau bisa disebut juga sebagai perusahaan Badan Usaha Milik Negara (BUMN). Telkom memiliki 11 direktorat di antaranya adalah Direktorat Digital Business (DDB). Di DDB, penulis mendapatkan kesempatan magang untuk menjadi *tester* di salah satu tim *Tribe IOT Platform* bernama IndiCar. IndiCar adalah sebuah aplikasi berbasis *Internet of Things* (IoT) yang berguna untuk monitor telemetri suatu kendaraan dengan memasang beberapa sensor di kendaraan. Tugas penulis sebagai seorang *tester* di tim IndiCar yaitu menguji website secara *end-to-end*. Proses pengujian *end-to-end* dapat dilakukan dengan dua cara yaitu secara manual maupun secara otomatis. Akan tetapi, website IndiCar mempunyai skala sistem yang besar dan membutuhkan waktu yang sangat lama apabila dilakukan secara manual. Hal ini dapat dicegah dengan menguji sistem secara otomatis menggunakan perkakas gratis bernama Cypress.

Cypress memiliki fitur dashboard yang iteratif yang dapat membantu dalam melakukan pengecekan website dengan sangat cepat dan mudah. Tidak hanya itu, Cypress memiliki komunitas yang besar dan mudah digunakan. Adapun keunggulan Cypress yang sangat dirasakan oleh penulis yaitu dalam segi penyimpanan maupun penggunaan jaringan, Cypress sangat ringan dibandingkan perkakas sejenis. Dengan adanya keunggulan-keunggulan yang dimiliki oleh Cypress, pengujian *end-to-end* terhadap website IndiCar lebih mudah dilakukan karena mempunyai IndiCar mempunyai alur pengujian yang sangat banyak. Akan tetapi, Cypress memiliki kekurangan yaitu tergantung pada suatu elemen tertentu dan apabila elemen tersebut berubah maka terjadi kesalahan. Akan tetapi, penggunaanya tidak perlu khawatir karena Cypress juga memiliki komunitas yang luas sehingga permasalahan tersebut dapat terselesaikan dengan mencari di Google.

Kata kunci: Pengujian IoT, Pengujian *end-to-end*, *Automated testing*, Cypress, IndiCar.



## GLOSARIUM

<i>Black Box testing</i>	metode pengujian perangkat lunak untuk menguji fungsionalitas dari sebuah aplikasi
<i>End-to-end</i>	metode untuk memastikan bahwa aplikasi dapat berjalan dari awal hingga akhir dan mendapatkan hasil yang sesuai dengan yang diharapkan
<i>Automated testing</i>	Pengujian yang dilakukan secara otomatis dengan bantuan perkakas Cypress.
Cypress	perkakas pengujian <i>end-to-end</i> yang dilakukan secara otomatis untuk mempermudah pengujian <i>end-to-end</i> .
Debug	langkah untuk menelusuri kesalahan kode program.
GPS	alat untuk melacak posisi dengan bantuan <i>sim card</i> .
<i>Internet of Thing (IoT)</i>	alat yang ditanamkan berbagai sensor dan konektivitas untuk menyelesaikan suatu permasalahan.
<i>Agile</i>	metode pengembangan perangkat lunak yang dapat menyesuaikan dengan kebutuhan pelanggan.
<i>Scrum</i>	metode manajemen tim yang digunakan oleh tim dalam melakukan pengaturan secara mandiri.
<i>Sprint</i>	aktivitas yang harus diselesaikan saat menggunakan <i>scrum</i> .
<i>Telemetry</i>	sebuah teknologi untuk melakukan pengukuran secara jarak jauh.

## 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 .....	8
GLOSARIUM.....	9
DAFTAR ISI.....	10
DAFTAR TABEL.....	12
DAFTAR GAMBAR.....	13
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Ruang Lingkup.....	4
1.3. Tujuan .....	4
1.4. Manfaat .....	4
1.5. Sistematika Penulisan .....	5
<b>BAB II KAJIAN PUSTAKA.....</b>	<b>6</b>
2.1. Sistem <i>Internet of Things</i> (IoT).....	6
2.2. GPS .....	6
2.3. IndiCar .....	8
2.3.1. Proses Bisnis .....	9
2.3.2. Sistem Kerja.....	9
2.3.3. Server .....	10
2.4. <i>Agile</i> .....	10
2.5. Pengujian Sistem.....	11
2.5.1. Pengujian <i>End-to-End</i> .....	11
2.5.2. Pengujian Otomatis .....	12
2.5.3. Cypress.....	12
2.6. Migrasi Server.....	13
<b>BAB III PELAKSANAAN MAGANG.....</b>	<b>14</b>
3.1. Persiapan Magang.....	14
3.2. Manajemen Proyek .....	14
3.3. Pengujian Web Menggunakan Cypress .....	18
3.3.1. Pembuatan Skenario Pengujian.....	19
3.3.2. Perangkat Pengujian .....	19
3.3.3. Validasi Skenario Pengujian.....	20
3.3.4. Pembuatan Kode Cypress .....	21
3.3.1.4 Instalasi Cypress.....	21
3.3.2.4 Pembuatan Kode Cypress.....	22
3.3.3.4 Pembuatan Fail Kredensial.....	23
3.3.4.4 Menghubungkan Fail Pengujian dengan Fail Kredensial.....	24
3.3.5. Eksekusi Kode Pengujian.....	25
3.3.6. Hasil Eksekusi.....	35
<b>BAB IV REFLEKSI PELAKSANAAN MAGANG.....</b>	<b>36</b>
4.1. Relevansi Akademik .....	36

4.1.1. Pengujian <i>End-to-End</i> .....	36
4.1.2. Pengujian dengan Cypress .....	36
4.1.3. Perbandingan Cypress .....	37
4.1.4. Proses Migrasi Server .....	38
4.1.5. <i>Scrum</i> .....	39
4.2. Pembelajaran Magang.....	39
BAB V PENUTUP .....	41
5.1. Kesimpulan .....	41
5.2. Saran.....	41
DAFTAR PUSTAKA .....	42



**DAFTAR TABEL**

Tabel 3.1 Anggota tim IndiCar beserta <i>role</i> -nya .....	16
Tabel 3.2 Daftar <i>role</i> serta bagian di Trello.....	17
Tabel 3.3 Detail pengujian admin IndiCar.....	25
Tabel 3.4 Detail pengujian pada Webapp IndiCar.....	32
Tabel 4.1 Perbandingan Cypress dengan Perangkat Serupa .....	38



## DAFTAR GAMBAR

Gambar 1.1 Logo Telkom Indonesia .....	1
Gambar 1.2 <i>Onigram Digital Business Directorate</i> .....	2
Gambar 2.1 Data kasus pencurian dengan tindak kekerasan tahun 2016-2020.....	8
Gambar 2.2 Data kasus pencurian tanpa tindak kekerasan tahun 2016-2020.....	8
Gambar 3.1 Dokumentasi pelaksanaan <i>sprint planning</i> .....	15
Gambar 3.2 Dokumentasi pelaksanaan <i>daily sprint</i> .....	16
Gambar 3.3 Contoh penggunaan Google Sheets .....	19
Gambar 3.4 Aplikasi pengujian end-to-end (Cypress) .....	20
Gambar 3.5 Contoh Google Sheet yang telah divalidasi dan dilakukan pengujian.....	21
Gambar 3.6 Kode instalasi Cypress.....	22
Gambar 3.7 Instalasi dan proses menjalankan Cypress .....	22
Gambar 3.8 Tampilan list fail pengujian Cypress .....	23
Gambar 3.9 Contoh kode pengujian di Cypress .....	23
Gambar 3.10 <i>Script</i> fail JSON .....	24
Gambar 3.11 Contoh kode menggunakan objek yang ada di fail kredensial.....	24
Gambar 3.12 Hasil <i>test runner</i> Admin IndiCar .....	25
Gambar 3.13 Hasil test runner Webapp IndiCar.....	32
Gambar 3.14 Contoh tampilan kesalahan di <i>test-runner</i> Cypress .....	35
Gambar 3.15 Tampilan di terminal Cypress.....	35

## BAB I PENDAHULUAN

### 1.1. Latar Belakang

PT Telkom Indonesia (Persero) Tbk (Telkom) adalah perusahaan yang bergerak dibidang jasa layanan teknologi informasi dan komunikasi (TIK) dan jaringan telekomunikasi di Indonesia yang dimiliki oleh negara atau bisa disebut juga sebagai perusahaan Badan Usaha Milik Negara (BUMN). Untuk bertransformasi menjadi *digital telecommunication company*, TelkomGroup telah mengimplementasikan strategi bisnis dan operasional perusahaan yang berorientasi kepada pelanggan (*customer-oriented*). Transformasi tersebut akan membuat organisasi TelkomGroup menjadi lebih *lean* (ramping) dan *agile* (lincah) dalam perubahan industri telekomunikasi yang berlangsung sangat cepat. Hal tersebut bertujuan untuk “Mewujudkan bangsa yang lebih sejahtera dan berdaya saing serta memberikan nilai tambah yang terbaik bagi para pemangku kepentingan”. Maka dari itu, Telkom memiliki visi “Menjadi digital telco pilihan utama untuk memajukan masyarakat” dan salah satu misinya “Mempercepat pembangunan infrastruktur dan platform digital cerdas yang berkelanjutan, ekonomis, dan dapat diakses oleh seluruh masyarakat” (Telkom Indonesia, 2020).

Untuk menggambarkan tekad dalam mewujudkan visi dan misi dari sebuah perusahaan, maka perlu adanya logo yang menggambarkan ciri khas dari perusahaan. Telkom Indonesia sendiri memiliki sebuah logo perusahaan yang terdiri dari bentuk lingkaran yang melambangkan kedinamisan perusahaan dan juga simbol dari dunia, serta tangan kanan yang ramah dalam meraih dunia tersebut. Dari sisi warna, Telkom Indonesia memiliki kombinasi warna merah, putih, dan warna transisi abu-abu sebagai identitas perusahaan, sebagaimana warna merah dan putih merupakan identitas negara Indonesia. Detail logo dari PT Telkom Indonesia (Persero) Tbk sendiri dapat dilihat pada Gambar 1.1.

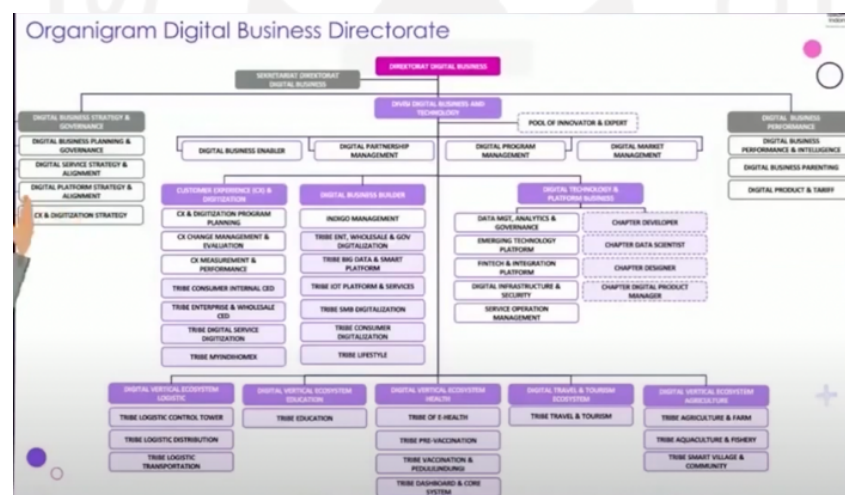


Gambar 1.1 Logo Telkom Indonesia

Sumber: *Website Resmi Telkom Indonesia*

Dengan adanya logo serta visi dan misi yang kuat, Telkom Indonesia telah mendapatkan banyak penghargaan. Penghargaan yang didapatkan oleh Telkom Indonesia berasal dari berbagai penyelenggara, seperti BUMN, itech, Top Business, TRAS N CO Indonesia, dan masih banyak lagi (Telkom Indonesia, 2022). Agar dapat terus berkembang dan terus memberikan layanan yang terbaik, PT Telkom mempunyai delapan direktorat, salah satunya adalah Direktorat Digital Business (DDB). Telkom DDB sendiri memiliki 11 kantor yang tersebar di Bandung dan di Jakarta (DDB Telkom, 2022). Lokasi kantor Telkom DDB yang ada di Bandung terletak di Jalan Setrasari Indah No 47, Sukarasa, Sukasari, Jawa Barat. Sedangkan kantor Telkom DDB yang ada di Jakarta berlokasi di Menara Multimedia lantai 15 Jalan Kebon Sirih No 12, Jakarta Pusat.

Telkom DDB juga merupakan divisi terpusat yang memiliki aktivitas bisnis yang bertujuan pada pengelolaan inovasi digital. Telkom DDB memiliki 3 divisi, salah satu dari 3 divisi tersebut memiliki 12 bagian. Agar dapat lebih jelas dapat dilihat pada Gambar 1.2 tentang *Onigram Digital Business Directorate*:



Gambar 1.2 *Onigram Digital Business Directorate*

Sumber: Video Pisah Sambut Telkom DDB

Penulis mendapatkan informasi magang di Telkom Indonesia melalui relasi kampus. Terdapat beberapa tahapan yang telah penulis lakukan seperti seleksi berkas dan wawancara. Akan tetapi untuk proses wawancara itu sendiri dibagi menjadi dua, yaitu membuat video perkenalan diri dan wawancara langsung bersama tim terkait. Setelah lolos seleksi, penulis juga akhirnya masuk ke dalam bagian *Digital Business Builder* yang lebih tepatnya pada *Tribe IOT Platform* khususnya pada tim IndiCar, KBM Online, dan PJU Online yang berlokasi di Bandung Digital Valley atau bisa disebut IndigoHub Bandung yang berlokasi di Digital Valley

lantai 2 Menara Bandung Jalan Gegerkalong Hilir No 47, Sukarasa, Sukasari, Kota Bandung, Jawa Barat.

IndiCar adalah layanan *Internet of Things* (IoT) untuk digitalisasi kendaraan dengan menyediakan konektivitas dan monitoring telemetri sehingga kendaraan dapat digunakan secara aman, nyaman, dan produktif (*secure-comfort-productive*), baik bagi pengemudi, penumpang, pemilik dan bahkan bagi kendaraan itu sendiri. Dalam kesempatan magang di Telkom Indonesia, penulis mendapatkan kesempatan untuk melakukan pengujian *Application Programming Interface* (API) menggunakan Postman, melakukan pengujian manual, dan membuat pengujian otomatis menggunakan Cypress dan Postman dengan durasi kurang lebih delapan bulan. Selanjutnya yaitu KBM Online, KBM Online adalah suatu aplikasi pemesanan kendaraan berbasis secara *online* dengan basis web dan *mobile* yang diperuntukkan pegawai Telkom Indonesia di seluruh Indonesia. KBM Online mempunyai integrasi dengan IndiCar untuk melakukan pengecekan lokasi kendaraan mobil yang sedang digunakan. Hal yang penulis lakukan dalam projek ini sama seperti tugas penulis tim IndiCar dengan durasi kurang lebih 2 bulan. Project yang terakhir penulis lakukan adalah sebagai *tester* di tim PJU Online dengan durasi kurang lebih 2 bulan dan masih sampai sampai sekarang.

Dikarenakan tugas yang didapatkan oleh penulis seputar pengujian *website* yang nantinya akan difokuskan pada pengujian otomatis untuk memastikan penggunaan *website* ini berjalan tanpa adanya kemungkinan kesalahan yang dilakukan oleh pengguna. Hal ini berfungsi untuk memungkinkan kesalahan ditemukan oleh pelanggan yang berujung pada kerugian di IndiCar. Metode yang penulis gunakan adalah metode pengujian *end-to-end* yang berguna untuk meminimalisir kesalahan yang ada. Metode ini dapat dilakukan dengan cara manual maupun otomatis, tapi untuk melakukan pengujian terhadap aplikasi yang besar apabila dikerjakan secara manual akan memakan waktu. Oleh karena itu, penulis memilih melakukan pengujian secara otomatis.

Perkakas yang digunakan untuk pengujian otomatis ini adalah Cypress. Penulis memilih ini karena setelah membandingkan antara perkakas serupa, perkakas pengujian otomatis yang memiliki kelebihan dibandingkan perkakas yang lain adalah Cypress. Kelebihan yang dimiliki oleh Cypress seperti menggunakan jaringan yang sedikit, memiliki *dashboard* yang iteratif, dan masih banyak lagi. Tim IndiCar meminta penulis untuk membuat kode pengujian otomatis untuk pertama kali.

Hadirnya penulis di tim IndiCar diharapkan dapat membuat kode pengujian secara otomatis menggunakan Cypress untuk mempercepat proses pengujian. Penulis juga diharapkan



dapat menerapkan pengujian *end-to-end* secara otomatis untuk tim yang masih melakukan pengujian secara manual. Selain itu, penggunaan Cypress ini diharapkan dapat menjadi langkah awal dari pengujian yang dilakukan oleh tim IndiCar di kemudian hari.

## 1.2. Ruang Lingkup

Penulis mendapatkan kesempatan untuk magang melalui program *Internship* DDB Telkom yang dikelola oleh ibu Sendylenvi Regia atau biasa dipanggil ibu Sedy. Apabila ada hal yang berkaitan dengan program ini dapat ditanyakan langsung melalui grup atau mengirimkan email ke *Internship* DDB Telkom yang diteruskan ke ibu Sedy.

Selama magang di Telkom DDB, penulis mendapatkan kesempatan bergabung dengan divisi *tribe IoT platform & services* yang disupervisi oleh Kusnassriyanto Saiful Bahri. Di Telkom sendiri memiliki banyak tim, salah satunya adalah IndiCar. Di tim IndiCar sendiri terdapat *product owner* atau bisa disebut orang yang mengendalikan produk ini mau menjadi apa, *project manager* atau bisa disebut orang yang mengatur proyek berjalan sesuai waktu dan biaya yang ditentukan, lalu supervisor penulis menjadi *scrum master*, lalu ada satu orang anak magang bagian UI / UX, tiga anak magang dan dua pegawai sebagai *back-end developer*, dua pegawai sebagai *front-end developer*, dan penulis sebagai *tester*. Sebagai seorang tester, penulis mendapatkan tugas seperti:

- a. Membuat skenario pengujian
- b. Memilih perkakas pengujian yang tepat
- c. Membuat pengujian otomatis menggunakan Cypress
- d. Melaporkan kesalahan yang ditemukan saat melakukan pengujian menggunakan Cypress

## 1.3. Tujuan

Tujuan yang diangkat sebagai bagian dari ruang lingkup magang ini adalah melakukan pengujian otomatis dengan menggunakan Cypress agar dapat memastikan kemungkinan kesalahan yang dilakukan oleh pengguna dapat dicegah.

## 1.4. Manfaat

Manfaat yang dapat diambil setelah adanya kode pengujian otomatis adalah mempercepat saat ada perbaikan kesalahan yang kecil. Tidak hanya itu, ada pula manfaat lainnya seperti mempersingkat waktu untuk bagian pengujian saat ada migrasi server. Penulis

mendapatkan kesempatan pengujian saat proses migrasi berlangsung yang membutuhkan waktu kurang lebih 30 menit, sedangkan pengujian dengan menggunakan Cypress hanya membutuhkan waktu kurang lebih 5 menit

### 1.5. Sistematika Penulisan

Dalam penyusunan laporan penelitian ini, sistematika penulisan dibagi menjadi beberapa lima bab, yaitu:

e. BAB I: Pendahuluan

Pada bab ini akan menjelaskan tentang pembahasan secara umum yang meliputi latar belakang, ruang lingkup, tujuan, manfaat dan sistematika penulis.

f. BAB II: Kajian Pustaka

Pada bab ini akan menjelaskan tentang pembahasan terkait teori sistem *Internet of Thing* (IoT) dan GPS. Selain itu, pada bab ini juga membahas teori tentang proses bisnis IndiCar, sistem kerja IndiCar, dan server yang ada di IndiCar. Kemudian terdapat teori yang membahas *agile*, pengujian sistem, dan migrasi server.

g. BAB III: Pelaksanaan Magang

Pada bab ini akan menjelaskan tentang manajemen pelaksanaan magang. Manajemen ini meliputi metode yang digunakan dalam pelaksanaan magang, perkakas yang digunakan, hingga cara penulisan kode pengujian otomatis.

h. BAB IV: Refleksi Pelaksanaan Magang

Pada bab ini akan menjelaskan tentang analisis terkait gap antara teori-teori/kajian akademik yang telah diuraikan di bab 2 dengan pelaksanaan di lapangan selama magang (empiris). Selain itu, terdapat pula hal-hal yang dipelajari selama magang berlangsung, khususnya yang diperoleh dari gap analysis pada sub bab sebelumnya.

i. BAB V: Kesimpulan

Pada bab ini akan menjelaskan tentang hasil kesimpulan berdasarkan BAB I- BAB V. Selain itu, pada bab ini terdapat mengecek bagian yang sesuai dengan tujuan awal, tercapainya tujuan awal dari magang ini, hingga saran yang diperlukan dalam penulisan ini.

## BAB II KAJIAN PUSTAKA

### 2.1. Sistem *Internet of Things* (IoT)

*Internet of Things* (IoT) adalah sebuah konsep yang berguna untuk memperluas keuntungan dari selalu terhubungnya ke internet untuk berbagi data, mengontrol sesuatu dari jarak jauh, dan masih banyak lagi (Rahmadhani dkk., 2022). Data yang dihasilkan berasal dari sensor yang ada di perangkat cerdas dan dikirimkan ke server. Selain itu, menurut Villamil beserta koleganya mengatakan bahwa *internet of things* (IoT) adalah sebuah perangkat keras yang dapat terhubung ke jaringan yang dapat berinteraksi tanpa campur tangan manusia. Hal ini dapat meningkatkan kualitas hidup manusia dan terdapatnya fasilitas untuk akses ke informasi serta layanan tertentu (Villamil dkk., 2020).

Terdapat berbagai contoh pengaplikasian sistem IoT yang ada di dunia saat ini, salah satunya adalah sistem pelacakan kendaraan. Ekky menyebutkan bahwa terdapat perangkat yang dapat melacak suhu kulkas yang memiliki sensor suhu di dalam kendaraan seperti truk. Apabila suhu di dalam kulkas tidak sesuai dengan yang diinginkan, maka akan mengirimkan notifikasi ke dalam website yang ada (Alam, 2020). Selain itu, dalam sistem pelacakan kendaraan perlu adanya sensor GPS yang terdapat di perangkatnya. Data lokasi kendaraan nantinya akan dikirimkan ke server dan dapat dilihat pada website yang telah disediakan. Hal ini berkaitan erat dengan IndiCar yang mana pengguna dapat melihat kendaraan yang telah dipasangkan perangkat IoT di kendaraannya melalui website IndiCar webapp.

### 2.2. GPS

Dalam melakukan sebuah pelacakan, teknologi yang paling sering digunakan adalah *Global Positioning System* (GPS). GPS yang dapat melacak posisi kendaraan hingga mencatat histori perjalanan. GPS ini banyak digunakan di berbagai aplikasi dari perusahaan berbeda, seperti Peduli Lindungi, Gojek, *drone*, TemanBus, dan masih banyak lagi. GPS pada umumnya ditanamkan pada berbagai jenis ponsel pintar salah satunya menggunakan teknologi GSM dan RFID (K. Moorthi dkk., 2018).

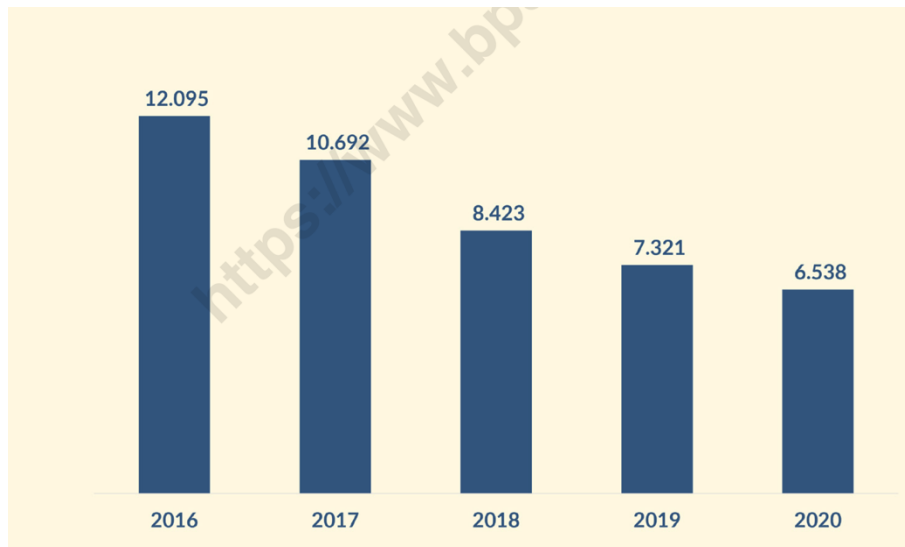
Pada aplikasi Peduli Lindungi, penggunaan GPS yang ada menggunakan sistem GPS yang ada pada ponsel pintar. Menurut Fadli, aplikasi Peduli Lindungi menggunakan GPS untuk melacak pergerakan penggunanya untuk memberi tahu keadaan sekitarnya untuk menjalani

protokol kesehatan apabila ada yang terpapar Covid-19 (Fadli, 2020). Hal ini dapat membantu para penggunanya untuk mengurangi terpaparnya Covid-19.

Sama halnya dengan aplikasi Peduli Lindungi, aplikasi ojek online seperti Gojek pun berasal dari ponsel pintar yang dikendarai oleh *driver* dan pelanggannya. Hal ini didukung dengan adanya penelitian Wahyuningtyas yang menyebutkan bahwa lokasi GPS yang ada pada *driver* berguna dalam memprioritaskan *driver* yang terdekat dari pelanggan dan adanya perkiraan waktu *driver* sampai atau pelanggan sampai ke tujuan (Wahyuningtyas, 2019).

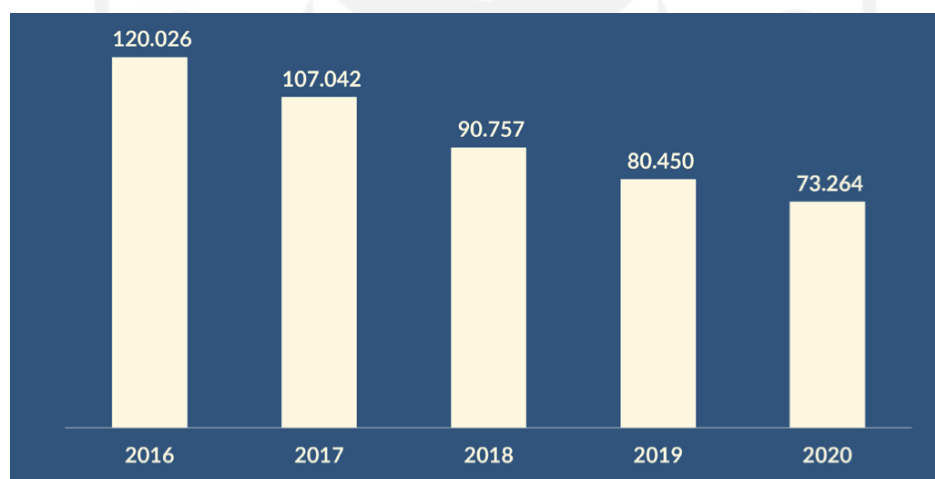
Berbeda dari sistem kerja dari aplikasi sebelumnya, sistem *tracking* pada *drone* yang menggunakan alat GPS yang langsung dipasangkan drone. Dalam hal ini Gowda beserta koleganya menyebutkan bahwa *drone* memiliki sistem GPS yang sangat banyak dikarenakan terdapat alasan yang dapat menghalang sinyal dari GPS itu sendiri (Gowda dkk., 2016). Dengan adanya sistem GPS yang banyak dapat meminimalisir kehilangan *drone* saat cuaca sedang buruk ataupun tersangkut di tempat yang tidak kelihatan.

Berbeda dari yang lain, proses *tracking* kendaraan yang diterapkan IndiCar adalah menggunakan alat yang nantinya dipasangkan di mobil. Sistem kerja dari alat IndiCar ini menggunakan bantuan Global System for Mobile Communications (GSM) untuk mengirimkan lokasi kendaraan. Menurut Hlaing bersama koleganya menyebutkan bahwa sistem GPS seperti ini akan adalah suatu cara yang efektif untuk melacak kendaraan secara *real-time* (Hlaing dkk., 2019). Dengan adanya hal ini, para pengguna IndiCar dapat lebih aman dari kasus pencurian yang ada di Indonesia karena kendaraan yang mereka miliki dilacak posisinya melalui GPS secara *real-time*. Berdasarkan data dari Polri kasus kejahatan terhadap hak/milik, pada tahun 2020 terdapat 79.802 kasus yang terdiri dari 6.538 kasus tindak pencurian dengan kekerasan dan 73.264 kasus tindak pencurian tanpa kekerasan (Badan Pusat Statistik, 2021). Agar dapat lebih jelas, dapat dilihat pada Gambar 2.1 Data kasus pencurian dengan tindak kekerasan tahun 2016-2020 dan Gambar 2.2.



Gambar 2.1 Data kasus pencurian dengan tindak kekerasan tahun 2016-2020

Sumber: Badan Pusat Statistik, 2021



Gambar 2.2 Data kasus pencurian tanpa tindak kekerasan tahun 2016-2020

Sumber: Badan Pusat Statistik, 2021

Berdasarkan jumlah kasus diatas, masih banyak kasus tindak pencurian yang dilakukan baik dengan kekerasan ataupun tanpa kekerasan. Akan tetapi, pengguna IndiCar tidak perlu takut dengan adanya pencurian karena pada kendaraan yang dimiliki oleh pelanggan telah dipasangkan GPS yang nantinya dapat ditemukan dengan mudah. Selain itu, dengan adanya IndiCar, diharapkan dapat menurunkan kasus pencurian yang ada di Indonesia.

### 2.3. IndiCar

IndiCar adalah sebuah sistem berbasis *Internet of Things* (IoT) yang sangat berguna untuk melacak kendaraan menggunakan sistem GPS, melacak kondisi kendaraan (RPM, suhu

mesin, indikator bensin, dan kecepatan saat berkendara), hingga melihat perilaku berkendara (Telkom Indonesia, 2022). IndiCar menyediakan jasa pembuatan layanan pelacakan kendaraan bagi perusahaan yang belum menggunakan alat pelacak di mobilnya atau sudah memiliki alat pelacak dari pihak lainnya. Saat ini, IndiCar telah mempunyai ribuan pengguna dari berbagai perusahaan, seperti Telkom Indonesia, Biofarma, dan masih banyak lagi.

### **2.3.1. Proses Bisnis**

Proses bisnis adalah sebuah dokumentasi dari sebuah aplikasi untuk mempermudah pemahaman terhadap aplikasi yang dibuat baik secara alur agar dapat mencapai sebuah tujuan (Sukmawati & Priyadi, 2019). IndiCar memiliki dua proses bisnis, yaitu melacak kendaraan yang menggunakan website IndiCar dan membuka jasa layanan pembuatan sistem bagi perusahaan untuk melacak kendaraan yang mereka miliki. Dalam pembuatan sebuah sistem perusahaan, IndiCar biasanya mengikuti sebuah tender yang mana nantinya apabila menang, maka akan dibuatkan sistem berdasarkan perjanjian yang ada. Aplikasi yang IndiCar bangun akan dirawat oleh tim IndiCar apabila terjadi suatu permasalahan di sistem yang telah dibuat.

### **2.3.2. Sistem Kerja**

Sistem kerja yang dimiliki oleh IndiCar adalah pelanggan akan didaftarkan ke dalam website admin IndiCar. Setelah itu, pelanggan akan menerima perangkat IoT yang sudah terpasang *sim card* di dalamnya dan akan dipasangkan oleh tim lapangan di IndiCar. Perangkat yang digunakan di IndiCar menggunakan sebuah perangkat IoT yang dipasang ke kendaraan pelanggan untuk mendapatkan data secara *real-time* bertujuan untuk memudahkan perusahaan melacak kendaraan apabila terjadi sesuatu hal yang tidak diinginkan. Perangkat ini telah merubah cara pengambilan data meskipun membutuhkan biaya yang lebih mahal. Hal ini juga didukung dengan adanya jurnal yang membahas tentang perangkat pelacak ini memiliki keuntungan substansial yaitu tidak terpengaruhnya dengan perilaku manusia, seperti mengirimkan data 24 jam tanpa henti (Rix dkk., 2022).

Apabila alat IndiCar sudah terpasang, maka pengguna dapat melihat posisi kendaraan yang dimiliki, riwayat perjalanan untuk masing-masing kendaraan, cara seseorang membawa kendaraan, skor berkendara, hingga tagihan yang mereka miliki di IndiCar webapp. Apabila pelanggan tersebut memiliki website diluar IndiCar, maka dapat melakukan permintaan untuk mengakses API dari IndiCar. Ketika permintaan tersebut disetujui, tim IndiCar akan membantu untuk menyiapkan dokumentasi API yang dibutuhkan.

### 2.3.3. Server

Dengan adanya pengiriman data lokasi dari alat yang dipasangkan ke kendaraan pengguna secara *real-time*, Telkom Indonesia membutuhkan penyimpanan dan pencadangan di server yang besar. Sayangnya, penyimpanan di server membutuhkan biaya yang sangat mahal. Sama seperti kasus IndiCar, Anggris dan koleganya membuat sistem menggunakan GPS yang dikirimkan secara langsung ke dalam *website* yang berguna untuk menemukan posisi rambu-rambu lalu lintas membutuhkan biaya server yang cukup besar (Anggris dkk., 2018). Untuk mengurangi biaya server tersebut, developer IndiCar memindahkan data yang tersimpan di penyimpanan server inti ke pencadangan server. Dengan proses yang dilakukan oleh developer, proses bisnis yang ada di IndiCar dapat terus lancar baik dari segi ekonomi maupun efektivitas aplikasi yang disebabkan data yang dikirimkan terus menerus dari semua kendaraan yang ada.

### 2.4. Agile

Dalam proses pengembangan suatu aplikasi terdapat beberapa metode, salah satunya *agile*. *Agile* adalah sebuah metode pengembangan aplikasi yang tangkas, ideal, dan paling tepat dalam pengembangan aplikasi (Govil & Sharma, 2022). Hal ini dibuktikan dalam pengembangan aplikasi di IndiCar karena memiliki fitur yang dapat berubah sesuai kebutuhan. Hal ini disebabkan oleh keinginan dari *stakeholder* yang berubah-ubah sesuai dengan kebutuhan.

Sedangkan menurut Lee dan Xia dalam jurnal Tassarolo F dan koleganya, *agile* adalah sebuah metode pengembangan aplikasi yang digunakan secara luas di bisnis yang berfungsi dalam meningkatkan kinerja pengembangan aplikasi, efisiensi, dan efektif dalam kebutuhan *stakeholder* dan perubahan yang sering terjadi (Tassarolo dkk., 2022). Hal ini diterapkan dalam pengembangan aplikasi IndiCar yang mana fitur yang dibuat pertama kali adalah fitur yang urgensinya lebih besar.

Tidak hanya itu, dalam penerapan *agile* pada tim IndiCar membutuhkan *scrum* yang sangat berguna dalam proses pengembangan aplikasi. *Scrum* adalah sebuah metode pengembangan aplikasi kolaborasi tim yang menggunakan metode *agile* untuk mencapai sebuah tujuan dari aplikasi itu dibangun (Firman Ashari dkk., 2022). Dari penjelasan *scrum* ini, proses pengembangan aplikasi IndiCar oleh tim dapat terpantau dengan jelas fitur apa saja yang sudah selesai dibangun dan fitur yang dalam proses pengembangan.



Menurut N. Naik dan P. J. Relax, *scrum* adalah metode pengembangan yang menggunakan pendekatan agile karena fleksibilitas dan transparansi dalam proses pengembangan (Alshammari, 2022). Dikarenakan fitur yang terus berubah sesuai dengan kebutuhan, maka diperlukan dokumentasi yang lengkap. Dalam proses pengembangan IndiCar, tim menggunakan Trello untuk pengembangan aplikasi. Trello sangat berguna untuk mencatat fitur yang akan dikembangkan serta melacak fitur itu sudah dikerjakan atau belum.

*Scrum* adalah sebuah metode yang dilakukan secara iteratif yang berguna untuk mendemonstrasikan produk ke pelanggan dengan menjalankan proses sprint (Jondya dkk., 2022). Berdasarkan penjelasan teori ini, dengan menjalankan metode *scrum*, maka perlu menjalankan proses *sprint*. Proses *sprint* diterapkan oleh tim IndiCar untuk menyelesaikan pengembangan aplikasi dan setelah selesai *sprint* akan didemokan untuk melihat fitur yang telah dibuat.

## 2.5. Pengujian Sistem

Untuk memastikan sebuah aplikasi berjalan sesuai rencana dan tanpa adanya kesalahan, perlu dilakukan pengujian. Pengujian adalah sebuah elemen kritis dalam menguji kualitas sebuah aplikasi (Snadhika Jaya, 2018). Pengujian biasanya dilakukan saat aplikasi sedang dibuat atau saat proses pengembangan serta saat proses migrasi server. Pengujian ini biasanya dilakukan oleh seseorang yang hanya melakukan pengujian, bukan dari orang yang membangun aplikasi. Dengan adanya hal ini, pengujian dapat berlangsung dengan objektif tanpa ada yang disembunyikan. Salah satu jenis pengujian yang sering dilakukan adalah pengujian *end-to-end*.

### 2.5.1. Pengujian *End-to-End*

Pengujian *end-to-end* yaitu urutan langkah/tindakan yang dilakukan pada aplikasi dari awal hingga akhir. Hal ini didukung dari penjelasan Taky tentang *end-to-end testing*, yaitu sebuah pengujian yang mengecek aplikasi di semua sistem agar sesuai dengan yang diharapkan (Taky, 2021). Pada saat proses pembuatan IndiCar berlangsung, penjelasan dari Taky terkait pengujian *end-to-end* sangat diterapkan agar tidak membuat para *stakeholder* kecewa dengan aplikasi yang telah dibuat.

Selain itu Huoponen juga mengartikan bahwa pengujian *end-to-end* adalah sebuah cara untuk memastikan komponen-komponen yang ada sesuai dengan yang diharapkan (Huoponen, 2020). Pada studi kasus IndiCar melakukan proses migrasi *server*, teori ini berguna untuk



memastikan bahwa aplikasi yang dibuat dapat berjalan sesuai yang telah dibuat tanpa ada pengaruh dari migrasi *server*.

Sedangkan menurut Badal dan Grünler, pengujian *end-to-end* dapat menguji aplikasi pada tingkat sistem yang bersesuaian dengan skenario penggunaan. Akan tetapi, mereka juga menyebutkan bahwa pengujian *end-to-end* ini lambat dan tidak dapat diandalkan. Tidak hanya itu, mereka juga mengatakan bahwa apabila terjadi kegagalan dalam pengujian, pengujian *end-to-end* tidak dapat mengidentifikasi tersebut (Badal & Grünler, 2021). Berdasarkan pengalaman penulis terhadap studi kasus IndiCar, penulis tidak setuju dengan teori ini dikarenakan dengan menggunakan perkakas Cypress, pengguna dapat mengidentifikasi permasalahan tersebut.

### 2.5.2. Pengujian Otomatis

Pengujian otomatis adalah sebuah pengujian yang dilakukan secara otomatis dengan menggunakan kode program tertentu (Arcuri, 2019). Kode tersebut dibuat kemudian dijalankan yang berguna untuk melakukan pengujian *end-to-end* dengan waktu yang singkat. Menurut pengujian yang telah dilakukan oleh Mahomed dan koleganya 29 pengujian yang telah mereka lakukan hasilnya memenuhi kriteria (Mahomed dkk., 2021). Dengan adanya hal ini, pengujian manual dapat tergantikan dengan pengujian secara otomatis dengan hasil yang sama dan waktu yang lebih singkat.

### 2.5.3. Cypress

Cypress adalah sebuah *framework end-to-end testing open source* yang berguna untuk melakukan pengujian secara otomatis. Dengan menggunakan Cypress, para penggunanya sangat diuntungkan dari segi waktu dan tenaga. Hal ini juga di disebutkan dalam skripsi Nguyen yang menyebutkan bahwa Cypress adalah sebuah *framework* terbaik untuk *developers* dalam *end-to-end testing* (Nguyen, 2022). Tidak hanya itu, penjelasan tentang Cypress ini juga diperkuat oleh Taky yang menjelaskan bahwa Cypress adalah sebuah perkakas *open source* gratis tanpa harus bayar sepeserpun yang berguna untuk menguji tampilan secara gratis yang memiliki visual dalam proses pengujiannya (Taky, 2021). Dengan adanya visual ini dapat *debug* secara cepat dan *screenshot* kegagalan dalam proses pengujian berlangsung.

Tidak hanya itu, fitur yang disediakan oleh Cypress sangat banyak seperti memuat ulang browser saat ada perubahan di kode, mengarahkan kursor sesuai dengan kode yang dibuat, melakukan pengujian secara asinkron untuk menunggu waktu tertentu untuk suatu elemen,

mengukur waktu saat menjalankan proses pengujian, dan masih banyak lagi. Alajärvi juga menyebutkan bahwa Cypress mempunyai sintak yang mudah dipelajari dan proses pengujian akan berakhir apabila elemen yang dicari tidak ditemukan (Alajävi, 2022). Dengan kemudahan yang diberikan oleh Cypress, developer sangat menghemat waktu, tenaga, serta pikiran dalam mencari kesalahan yang mungkin terjadi saat melakukan *end-to-end testing*.

## 2.6. Migrasi Server

Server adalah komputer yang menyediakan layanan untuk komputer klien dengan bantuan jaringan (Muslihah dkk., 2021). Dengan adanya server ini, semua orang yang ada di dunia dapat mengakses sebuah informasi dengan mudah. Menurut Ketaren beserta koleganya, terdapat tiga jenis server seperti server platform, server aplikasi, dan *mail server* (Ketaren & Barus, 2020). Ketiga jenis server memiliki kapasitas penyimpanan yang dapat disesuaikan dengan kebutuhan bisnis yang sedang dijalankan. Apabila kapasitas penyimpanan server sudah penuh, maka perlu melakukan migrasi server ke server yang lebih besar atau yang lebih canggih. Seperti yang dilakukan Syafriadi beserta koleganya, mereka melakukan migrasi server untuk meningkatkan kinerja dari pengguna server (Syafriadi dkk., 2021).

Sebelum melakukan migrasi server, perlu mengidentifikasi terlebih dahulu kebutuhan server yang sesuai dengan aplikasi yang sedang dibangun, kemudian mendesain ulang topologi yang ada di server fisik (Wijayanto dkk., 2021). Telkom Indonesia melakukan proses migrasi yang diikuti oleh *product owner* untuk berkoordinasi dengan tim lainnya, *scrum master* untuk mengidentifikasi hasil yang didapatkan oleh developer, dan developer bertujuan untuk memilih spesifikasi server agar tidak salah dan menyebabkan kerugian di masa yang akan datang.

## BAB III

### PELAKSANAAN MAGANG

#### 3.1. Persiapan Magang

Setelah adanya pengumuman yang lolos magang pada bulan Desember 2021, semua peserta yang berhasil lolos magang di Telkom Indonesia melalui program DDB Internship Telkom akan diundang dalam acara pisah sambut program magang mahasiswa bersertifikat forum human capital Indonesia badan usaha milik negara (PMMB FHCI BUMN) Telkom Januari 2022. Dalam acara tersebut, penulis beserta kolega magang yang lolos akan melakukan perpisahan pada *batch* sebelumnya dan mendengarkan penjelasan seputar Telkom Indonesia.

Setelah acara tersebut selesai, penulis beserta peserta yang lolos magang dipisah berdasarkan supervisor yang ditetapkan. Pada kesempatan ini penulis disupervisi oleh Kusnassriyanto Saiful Bahri sebagai *scrum master* di tim IndiCar. Pada pertemuan, supervisor berkenalan dengan tim dan menjelaskan keahlian dari masing-masing individu yang berguna untuk penempatan posisi kerja beserta penetapan timnya.

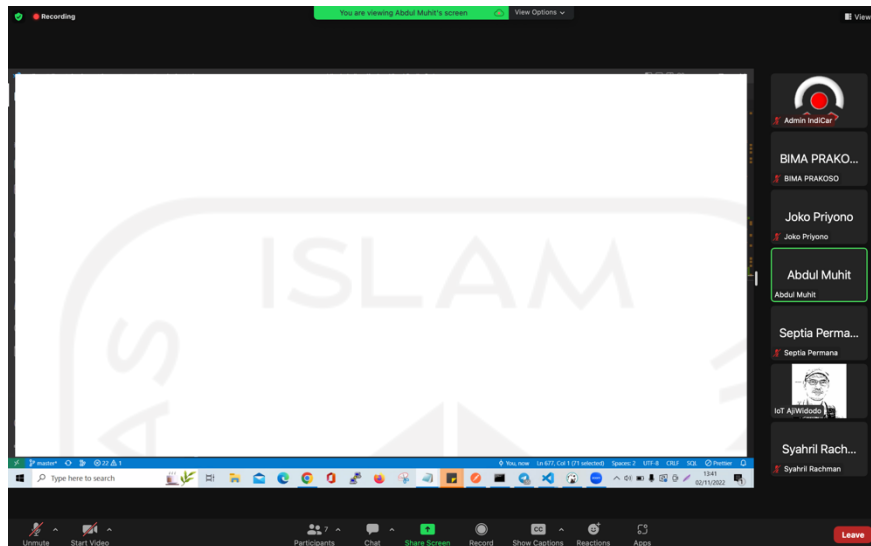
Pada kesempatan ini, penulis mendapat *job role* developer sebagai *quality assurance* (QA) di tim IndiCar yang tugas utamanya adalah memastikan aplikasi yang dibuat dapat berjalan dengan lancar serta cepat dalam pengujiannya. Oleh karena itu, penulis memilih Cypress sebagai perkakas yang membantu dalam proses pengujian *end-to-end*. Proses pemahaman seputar Cypress sangat mudah dikarenakan dokumentasi yang lengkap serta banyak orang yang sudah menggunakannya. Dengan menggunakan Cypress, penggunaannya harus menulis kode terlebih dahulu yang dilengkapi dengan bantuan untuk mendapatkan elemen yang diinginkan.

#### 3.2. Manajemen Proyek

Selama proses pembuatan aplikasi IndiCar berlangsung, metode manajemen proyek yang digunakan adalah *scrum*. Metode *scrum* ini adalah suatu metode manajemen proyek yang dilakukan selama dua minggu yang terdiri dari 4 aktivitas, yaitu *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective*.

Pada *sprint planning*, tim developer bersama *scrum master* melakukan pembahasan fitur apa yang akan dikerjakan dalam waktu 2 minggu ke depan. Setelah fitur-fitur tersebut ditulis ke dalam *backlog*, kami para developer melakukan penilaian yang terdiri dari beberapa poin, yaitu 1, 3, 5, 8, dan yang paling besar adalah 13. Poin tersebut akan dilihat yang diperoleh di setiap fitur nantinya akan dijumlahkan dan dibandingkan dengan *sprint* sebelumnya. Apabila

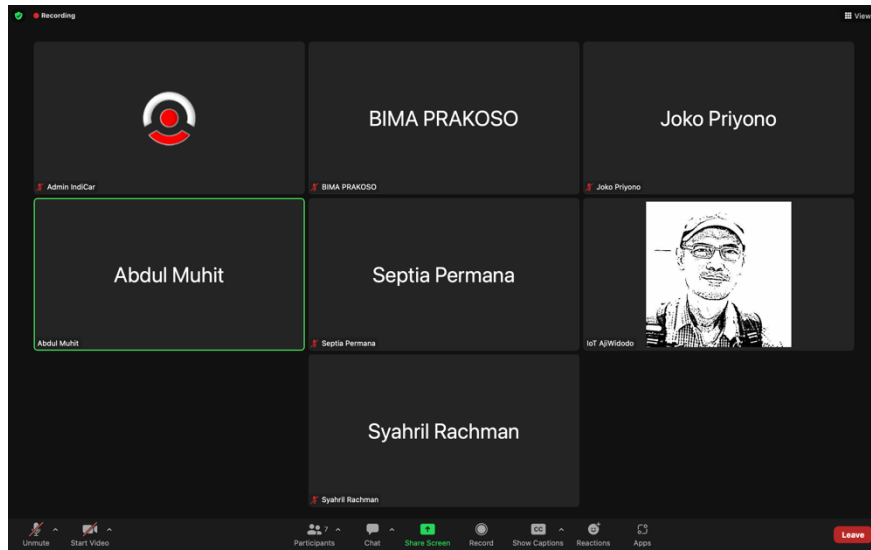
tidak ada poin pada *sprint* sebelumnya, maka poin yang dilakukan pada *sprint* sekarang akan digunakan untuk perbandingan di *sprint planning* berikutnya. Agar dapat lebih jelas, proses *sprint planning* yang dilakukan bersama developer lainnya dan *scrum master* pada Gambar 3.1.



Gambar 3.1 Dokumentasi pelaksanaan *sprint planning*

Selanjutnya, *daily scrum* adalah agenda lima belas menit setiap harinya yang berguna untuk melaporkan hasil pekerjaan yang kemarin dilakukan oleh semua anggota tim. Tidak hanya itu, *daily scrum* juga memiliki fungsi lain seperti apabila ada pekerjaan yang tidak dipahami maka dapat menanyakan langsung kepada orang yang mengerjakan tugas itu sebelumnya.

Berikutnya terdapat *sprint review*, *sprint review* adalah sebuah pelaporan hasil anggota tim kepada *scrum master* dan juga *product owner*. Pada kegiatan ini, *scrum master* akan mengecek pekerjaan yang telah dibuat oleh setiap anggota developer apakah fitur yang diharapkan sudah sesuai dengan yang dibuat. Selain itu, *scrum master* akan mengecek *backlog* mana saja yang sudah selesai dikerjakan. Apabila terdapat *backlog* yang belum selesai dikerjakan, *backlog* tersebut akan dilanjutkan pada *sprint* berikutnya. Akan tetapi, apabila *backlog* yang sudah selesai akan didemokan langsung oleh developer kepada *product owner* yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Dokumentasi pelaksanaan *daily sprint*

Setelah *sprint review* selesai, maka akan dilaksanakan *sprint retrospective*. *Sprint retrospective* adalah sebuah kegiatan yang berkaitan untuk mengintrospeksi diri selama sprint sebelumnya berlangsung. Dalam *sprint retrospective* ini, setiap orang menyebutkan kekurangan dirinya terlebih dahulu, lalu akan menyebutkan kekurangan dari anggota tim, perkakas yang digunakan, serta kekurangan dari proses *sprint* sebelumnya. Dengan adanya *sprint retrospective* ini diharapkan dapat membantu tim dalam memperbaiki kekurangan yang telah dilakukan pada sprint sebelumnya.

Dalam proses dokumentasi, Telkom Indonesia menggunakan perkakas bernama Trello yang dapat membantu tim serta *scrum master* dan *product owner* untuk *tracking* progress yang telah dibuat. Di dalam Trello, para penggunanya dapat membuat bagian-bagian serta card yang berisi tugas yang harus dikerjakan. Pada Tabel 3.1 Anggota tim IndiCar beserta *role*-nya dapat dilihat daftar anggota tim dari IndiCar beserta role dari setiap orang.

Tabel 3.1 Anggota tim IndiCar beserta *role*-nya

Nama	Role Pekerjaan	Deskripsi
Aji Widodo	<i>Project Manager</i>	Bertanggung jawab atas tim dan memberikan solusi bisnis terhadap pihak eksternal.
Hafizh Arsyari	<i>Product Owner</i>	Bertanggung jawab dalam menerjemahkan keinginan project manager ke dalam fitur.
Kusnassriyanto Saiful Bahri	<i>Scrum Master</i>	Bertanggung jawab atas berjalannya proses pembuatan aplikasi IndiCar dalam tim.

Abdul Muhit	<i>Programmer Back-end</i>	Membuat aplikasi yang menyediakan API serta membuat <i>unit test</i> .
Septi Permana	<i>Programmer Back-end</i>	Membuat aplikasi yang menyediakan API serta membuat <i>unit test</i> .
Jhoko Priyono	<i>Programmer Front-end</i>	Membuat tampilan sesuai dengan tampilan dari desain yang sudah ada
Syahril Rachman	<i>Programmer Front-end</i>	Membuat tampilan sesuai dengan tampilan dari desain yang sudah ada
Sarah Navianti	<i>UI/UX Designer Intern</i>	Membuat desain sesuai dengan keinginan <i>project manager</i>
Bima Prakoso	<i>Quality Assurance (QA) Intern</i>	Melakukan pengujian sistem, memastikan semua fungsi berjalan dengan baik, dan membuat kode pengujian otomatis

Berdasarkan *role* yang ada pada Tabel 3.1, tidak semua *role* mempunyai bagian-bagian *card* yang ada di Trello, salah satunya yang tidak memiliki bagian adalah *project manager*. Agar lebih jelas bagian-bagian yang ada di Trello yang ada di tim IndiCar, dapat dilihat pada Tabel 3.2.

Tabel 3.2 Daftar *role* serta bagian di Trello

<b>Role Pekerjaan</b>	<b>Nama Bagian</b>	<b>Keterangan</b>
<i>Product Owner</i>	<i>IndiCar Product Backlog</i>	Pada bagian ini, <i>product owner</i> memberikan fitur apa saja yang ingin dikembangkan.
<i>Scrum Master</i>	<i>Sprint Done</i>	Pada bagian ini, <i>sprint</i> yang sudah berjalan akan dicatat dan dilengkapi dengan poin dari tingkat kesulitan hingga tanggal <i>sprint</i> itu berlangsung.
	<i>Sprint Backlog</i>	Pada bagian <i>sprint backlog</i> , <i>scrum master</i> akan merumuskan fitur yang akan dikerjakan saat <i>sprint</i> dan poin dari tingkat kesulitan fitur.
<i>Programmer &amp; UI/UX Designer Intern</i>	<i>Detail Sprint Backlog</i>	Programmer atau UI/UX designer akan merumuskan detail dari setiap <i>card</i> yang ada di <i>sprint backlog</i> .
	<i>ToDo</i>	Setelah dirumuskan, <i>programmer</i> ataupun UI/UX designer akan memisahkan dari detail <i>sprint backlog</i> menjadi bagian yang lebih kecil.
	<i>In Development</i>	Pada bagian ini, baik programmer ataupun

		UI/UX designer akan memindahkan <i>card</i> dari bagian <i>ToDo</i> ke <i>In Development</i> apabila sudah mulai dikerjakan
	<i>Blocked</i>	Pada bagian ini, kartu yang sudah dibuat dan mengalami kendala teknis akan diletakkan ke bagian ini hingga masalah terselesaikan
	<i>Code Review</i>	Pada bagian ini dikhususkan untuk <i>programmer</i> , kode program yang sudah dibuat akan dinilai. Apabila sekiranya <i>programmer</i> sudah <i>me-review</i> kode yang sudah dibuat, maka akan diajukan ke <i>scrum master</i> untuk <i>merge</i> di <i>Playcourt</i>
<i>Quality Assurance Intern</i>	<i>Waiting for QA</i>	Setelah proses <i>merge</i> selesai, <i>card</i> akan diletakkan ke <i>waiting for QA</i> untuk menunggu pengujian
	<i>In QA</i>	Apabila <i>card</i> akan dikerjakan, maka <i>card</i> dari <i>waiting for QA</i> akan dipindahkan ke <i>In QA</i> untuk proses pengujian
	<i>Rejected</i>	Pada bagian ini, kode program yang telah diuji mengalami masalah akan dipindahkan dari bagian <i>In QA</i> ke bagian ini
	<i>Done</i>	Apabila pengujian tidak ada masalah, maka <i>card</i> dari bagian <i>In QA</i> akan dipindahkan ke bagian <i>Done</i> untuk memberitahu bahwa sudah dilakukan pengujian tanpa ada masalah

Selain dari bagian yang ada di Trello, penulis sebagai QA juga dituntut untuk mampu membuat pengujian otomatis menggunakan Cypress di setiap *website* yang dikembangkan. Dikarenakan QA di tim IndiCar hanya penulis seorang diri, maka penulis dituntut untuk belajar Cypress hingga mengimplementasikan ke *website* yang dibuat.

### 3.3. Pengujian Web Menggunakan Cypress

Pada pengujian *website* IndiCar yang penulis lakukan menggunakan perkakas bernama Cypress. *Website* IndiCar sendiri terbagi menjadi 3 jenis yang terdiri dari *website* admin, *website* webapp, dan *website* CS IndiCar. Setiap jenis *website* terdapat 1 kode pengujian otomatis. Sampai penulis membuat tugas akhir ini, penulis sudah membuat kode pengujian otomatis di setiap jenis *website* dan terus diperbarui.



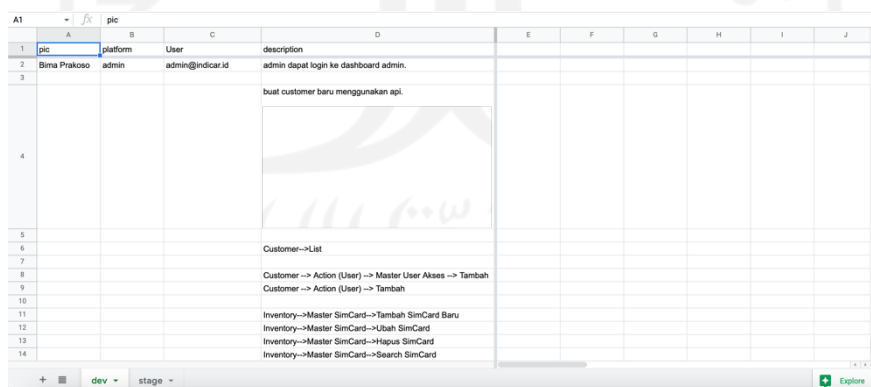
### 3.3.1. Pembuatan Skenario Pengujian

Sebelum membuat *script* pengujian, perlu adanya skenario terlebih dahulu agar tidak ada fitur yang terlewatkan. Di dalam skenario tersebut terdapat fitur apa saja yang akan diuji, hasil yang diharapkan, dan hasil pengujian. Tidak hanya itu, apabila menemukan bug saat proses pengujian dapat langsung ditulis ke dalam skenario tersebut. Selama proses pembuatan skenario pengujian, dibutuhkan analisis terlebih dahulu terhadap *website* yang ingin dibuatkan pengujiannya dan dituliskan di bagian deskripsi. Setelah semua selesai, sheets yang sudah dibuat akan divalidasi oleh *programmer front-end* dan *scrum master*. Pada *website* admin IndiCar terdapat 28 skenario dan pada *website* webapp IndiCar terdapat 91 skenario.

Skenario tersebut didapatkan dengan cara membuat list fitur yang telah tersedia. Kemudian, penulis melakukan pengujian secara manual dan mencatat hasil yang didapatkan. Semua tahapan serta hasil yang didapatkan dicatat yang nantinya akan menjadi panduan untuk membuat pengujian otomatis.

### 3.3.2. Perangkat Pengujian

Selama magang berlangsung, penulis menggunakan beberapa perangkat dalam pembuatan kode pengujian, yaitu Google Sheets dan Cypress. Google Sheets digunakan untuk menulis skenario pengujian yang mana skenario tersebut dapat dilihat oleh *scrum master* dan developer seperti pada Gambar 3.3.



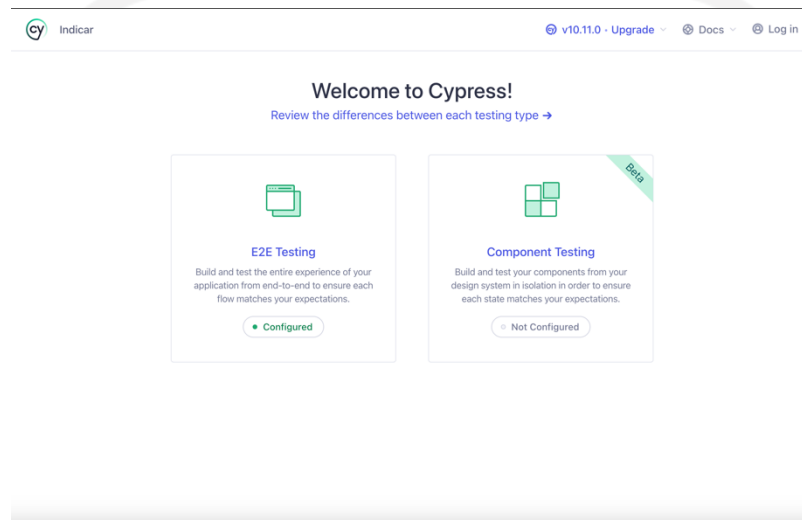
	A	B	C	D	E	F	G	H	I	J
1	plc	platform	User	description						
2	Bima Prakoso	admin	admin@indicar.id	admin dapat login ke dashboard admin.						
3				buat customer baru menggunakan api.						
4										
5										
6				Customer-->List						
7										
8				Customer --> Action (User) --> Master User Akses --> Tambah						
9				Customer --> Action (User) --> Tambah						
10										
11				Inventory-->Master SimCard-->Tambah SimCard Baru						
12				Inventory-->Master SimCard-->Ubah SimCard						
13				Inventory-->Master SimCard-->Hapus SimCard						
14				Inventory-->Master SimCard-->Search SimCard						

Gambar 3.3 Contoh penggunaan Google Sheets

Sedangkan Cypress digunakan untuk melakukan pengujian *end-to-end* secara otomatis. Penulis memilih Cypress karena Cypress adalah perangkat pengujian *end-to-end* yang gratis tanpa ada batasan penggunaan. Selain itu, Cypress memiliki komunitas yang besar sehingga apabila terjadi kesalahan dalam pembuatan kode pengujian maka dapat mencari solusinya dari internet.



Tidak hanya itu, Cypress memberikan pengalaman penggunanya yang baik karena memiliki *dashboard* yang iteratif untuk melihat proses pengujian yang sedang berlangsung. Cypress memiliki fitur yang lengkap diantara perkakas lainnya, seperti mengunggah gambar, video, ataupun melakukan pengujian API. Cypress termasuk perkakas pengujian yang ringan baik secara penyimpanan ataupun penggunaan jaringan, sehingga pengguna Cypress dapat menjalankan kode pengujian walaupun dalam kondisi sinyal yang stabil. Penggunaan Cypress dapat dilihat pada Gambar 3.4.



Gambar 3.4 Aplikasi pengujian end-to-end (Cypress)

### 3.3.3. Validasi Skenario Pengujian

Setelah skenario selesai dibuat ke Google Sheets, developer akan melihat apakah ada fitur yang kelewat untuk diuji atau tidak. Apabila semuanya telah sesuai, maka Google Sheets tersebut diberikan kepada *scrum master* dan *product owner* untuk divalidasi. Apabila list ini sudah divalidasi oleh *scrum master* dan *product owner*, maka dapat langsung digunakan untuk pengujian seperti Gambar 3.5. List fitur inilah yang nantinya akan digunakan untuk proses migrasi server ataupun pengujian saat perbaikan bug agar menghemat waktu.

	A	B	C	D	E	F	G	H	I	J
1	pic	platform	User	description	1 Agustus 2022	Catatan Bug	05/08/2022			
2	abdul muhit	admin	admin@indicar.id	admin dapat login ke dashboard admin.	v		v			
3				buat customer baru menggunakan api.						
4										
5										
6				Customer-->List	v		v			
7										
8				Customer --> Action (User) --> Master User Akses --> Tambah	v		v			
9				Customer --> Action (User) --> Tambah	v		v			
10										
11				Inventory-->Master SimCard-->Tambah SimCard Baru	v		v			
12				Inventory-->Master SimCard-->Ubah SimCard	v		v			
13				Inventory-->Master SimCard-->Hapus SimCard	v		v			
14				Inventory-->Master SimCard-->Search SimCard	v		v			

Gambar 3.5 Contoh Google Sheet yang telah divalidasi dan dilakukan pengujian

### 3.3.4. Pembuatan Kode Cypress

Setelah mendapatkan perkakas pengujian yang tepat, maka akan diuji secara manual terlebih dahulu untuk mengetahui hasil yang didapatkan secara langsung. Setelah mengetahui gambaran terhadap hasil yang akan didapatkan, maka perlu melakukan instalasi Cypress, pembuatan kode pengujian, dan pembuatan file kredensial.

#### 3.3.1.4 Instalasi Cypress

Cypress adalah salah satu aplikasi pengujian web yang ada di NodeJS. Selama magang berlangsung, penulis membuat kode Cypress dengan bantuan dokumentasi yang tersedia di website resmi Cypress. Dalam proses instalasi dan menjalankan Cypress, penulis menggunakan kode yang dijalankan di *Command Prompt (CMD)/Terminal* seperti pada Gambar 3.6 dan hasilnya dapat dilihat pada Gambar 3.7.

```
# Membuat folder Cypress
mkdir Indicar

# Masuk ke dalam folder Indicar
cd Indicar

# Memasang node modules di folder yang baru dibuat
npm init -y

# Memasang Cypress
```

```
npm install cypress --save-dev
```

Gambar 3.6 Kode instalasi Cypress.

```
bimaprakoso@MacBook-Pro-Bima magang % mkdir Indicar
bimaprakoso@MacBook-Pro-Bima magang % cd Indicar
bimaprakoso@MacBook-Pro-Bima Indicar % npm init --y
Wrote to /Users/bimaprakoso/Documents/Magang/Indicar/package.json:

{
  "name": "indicar",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

bimaprakoso@MacBook-Pro-Bima Indicar % npm install cypress --save-dev
added 165 packages, and audited 166 packages in 56s

28 packages are looking for funding
  run `npm fund` for details

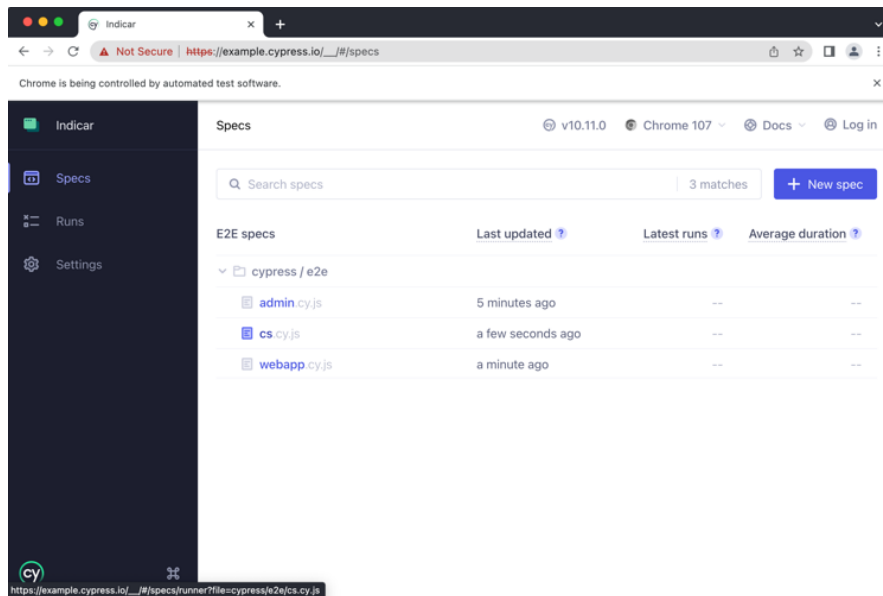
found 0 vulnerabilities
```

Gambar 3.7 Instalasi dan proses menjalankan Cypress

Pastikan hasil yang didapatkan di *Command Prompt (CMD)/Terminal* seperti Gambar 3.7. Apabila terjadi perbedaan, silahkan ulangi tahapan yang ada di Gambar 3.6. Setelah instalasi Cypress telah berhasil, silahkan jalankan Cypress dengan perintah “`npm run cypress:open`” untuk membuat kode pengujian.

#### 3.3.2.4 Pembuatan Kode Cypress

Setelah Cypress berhasil dijalankan, penulis membuat kode awal dengan cara mengklik E2E Testing, pilih browser yang digunakan, dan klik “*Create new empty spec*”. Penulis memberikan tiga nama fail yaitu admin, webapp, dan cs. Setelah semua fail selesai dibuat maka akan seperti Gambar 3.8 Tampilan list fail pengujian Cypress yang mana di setiap fail berisi seperti Gambar 3.9 Contoh kode pengujian di Cypress.



Gambar 3.8 Tampilan list fail pengujian Cypress

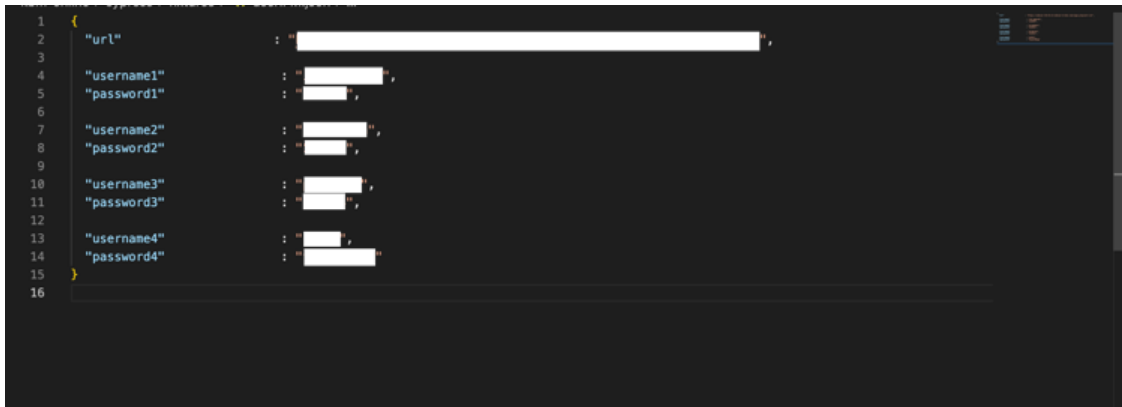
```
describe('empty spec', () => {
  it('passes', () => {
    cy.visit('https://example.cypress.io')
  })
})
```

Gambar 3.9 Contoh kode pengujian di Cypress

Selanjutnya, buka folder Cypress yang telah dibuat ke workspace di Visual Studio Code. Workspace di Visual Studio Code akan menampilkan folder dan fail yang ada di folder Cypress secara urut berdasarkan nama. Apabila ingin mengubah kode pengujian, silahkan buka folder cypress lalu pilih folder e2e dan pilih nama fail pengujian.

#### 3.3.3.4 Pembuatan Fail Kredensial

Tidak hanya itu, penulis menambahkan satu fail JSON (*JavaScript Object Notation*) yang berisi hal-hal yang akan digunakan kembali seperti URL web di dalam folder fixtures bernama “webapp.json”. Data yang ada di fail JSON tersebut dan dapat digunakan kembali dengan cara memanggil fail tersebut. Agar dapat lebih jelas seputar penulisan fail JSON yang ada di folder fixtures, dapat dilihat pada Gambar 3.10.

Gambar 3.10 *Script* fail JSON

### 3.3.4.4 Menghubungkan Fail Pengujian dengan Fail Kredensial

Setelah berhasil membuat fail kredensial, maka perlu menghubungkannya dengan fail pengujian. Hal pertama yang harus dilakukan adalah menambahkan tiga baris perintah. Kemudian, untuk memanggil sebuah objek dibutuhkan beberapa perintah tambahan. Agar dapat lebih jelas, dapat dilihat pada Gambar 3.11.

```

describe('empty spec', () => {
  // Memanggil file webapp.json
  beforeEach(() => {
    cy.fixture('webapp.json').as('user')
  })

  // Memanggil objek yang ada di fail kredensial
  before(() => {
    cy.get('@user').then((user) => {
      cy.visit(user.url)
      cy.get('[name="email"]').type(user.username1)
      cy.get('[name="password"]').type(user.password1)
    })
  })

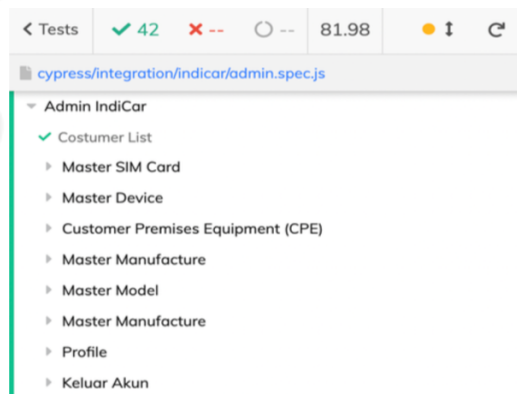
  it('passes', () => {
    cy.visit('https://example.cypress.io')
  })
})

```

Gambar 3.11 Contoh kode menggunakan objek yang ada di fail kredensial

### 3.3.5. Eksekusi Kode Pengujian

Dalam pengekseskuan *script test* menggunakan Cypress akan menghasilkan dua status pengujian, yaitu *passed* dan *CypressError* yang akan ditampilkan pada status halaman antarmuka yang ada. Status *passed* akan didapatkan apabila kode Cypress yang dibuat dapat dilakukan tanpa ada kesalahan sedikitpun. Agar dapat lebih jelas tentang hasil yang didapatkan *passed* dapat dilihat pada Gambar 3.12 .



Gambar 3.12 Hasil *test runner* Admin IndiCar

Pengujian yang dilakukan pada *website* admin IndiCar terdiri dari 8 kategori dengan 42 pengujian. Saat menjalankan proses pengujian di *website* admin IndiCar, membutuhkan waktu 81,98 detik. Detail dari pengujian yang dilakukan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Detail pengujian admin IndiCar

No	Kategori	Skenario Uji	Langkah-langkah	Ekspektasi	Hasil Realita	Status
1	Autentikasi	Login	1. Buka <i>website</i> admin IndiCar 2. Masukkan email dan <i>password</i> 3. Klik login	Pengguna dapat masuk ke akun	Sesuai dengan ekspektasi	Lolos
2	Customer	Melihat pelanggan di <i>Customer list</i>	1. Pengguna membuka halaman <i>customer</i>	<i>Website</i> memunculkan <i>list customer</i>	Sesuai dengan ekspektasi	Lolos
3		Mencari pelanggan di <i>customer list</i>	1. Masukkan <i>test@test.test</i> di formulir pencarian	<i>Website</i> menampilkan pelanggan dengan email <i>test@test.test</i>	Sesuai dengan ekspektasi	Lolos

4		Membuat akun pelanggan di <i>customer list</i>	1. Pengguna memasukkan data sesuai dengan form yang ada 2. Pengguna menekan tombol buat	<i>Website</i> dapat menyimpan data sesuai dengan data yang dimasukkan	Sesuai dengan ekspektasi	Lolos
5		Mengubah data pelanggan di <i>customer list</i>	1. Pengguna mencari pelanggan yang ingin diubah 2. Pengguna memperbarui data yang ada 3. Pengguna menekan tombol simpan	<i>Website</i> dapat menyimpan data yang baru diubah	Sesuai dengan ekspektasi	Lolos
7		Menghapus pelanggan di <i>customer list</i>	1. Pengguna mencari data yang ingin dihapus 2. Pengguna menekan tombol hapus	<i>Website</i> menghapus data yang ingin dihapus	Sesuai dengan ekspektasi	Lolos
8	<i>Sim Card</i>	Melihat <i>sim card</i> di <i>master sim card</i>	1. Pengguna mengklik halaman <i>sim card</i>	<i>Website</i> menampilkan <i>list sim card</i>	Sesuai dengan ekspektasi	Lolos
9		Mencari <i>sim card</i> di <i>Master Sim card</i>	1. Pengguna memasukkan nomor <i>sim card</i> di formulir pencarian	<i>Website</i> menampilkan <i>list sim card</i> dengan nomer <i>sim card</i> yang dimasukkan	Sesuai dengan ekspektasi	Lolos
10		Mencari <i>sim card</i> di <i>Master Sim card</i>	1. Pengguna memasukkan kode <i>sim card</i> di formulir pencarian	<i>Website</i> menampilkan <i>list sim card</i> dengan kode <i>sim card</i> yang dimasukkan	Sesuai dengan ekspektasi	Lolos
11		Membuat <i>sim card</i> di <i>master sim card</i>	1. Pengguna memasukkan data sesuai dengan formulir yang ada 2. Pengguna menekan tombol buat	<i>Website</i> dapat menyimpan data sesuai dengan data yang dimasukkan	Sesuai dengan ekspektasi	Lolos
12		Mengubah <i>sim card</i> di	1. Pengguna mencari <i>sim</i>	<i>Website</i> dapat menyimpan	Sesuai dengan ekspektasi	Lolos

		<i>master sim card</i>	<i>card</i> yang ingin diubah 2. Pengguna memperbaiki data yang ada 3. Pengguna menekan tombol simpan	data yang baru diubah		
13		Menghapus <i>sim card</i> di <i>master sim card</i>	1. Pengguna mencari data yang ingin dihapus 2. Pengguna menekan tombol hapus	<i>Website</i> menghapus data yang ingin dihapus	Sesuai dengan ekspektasi	Lolos
14	<i>Device</i>	Mengalihkan ke halaman <i>master device</i>	1. Pengguna mengklik halaman <i>device</i>	<i>Website</i> menampilkan <i>list device</i>	Sesuai dengan ekspektasi	Lolos
15		Mencari perangkat di <i>master device</i>	1. Pengguna memasukkan <i>imei</i> perangkat di formulir pencarian	<i>Website</i> menampilkan <i>list device</i> dengan <i>imei</i> perangkat yang dimasukkan	Sesuai dengan ekspektasi	Lolos
16		Mencari perangkat di <i>master device</i>	1. Pengguna memasukkan nama perangkat di formulir pencarian	<i>Website</i> menampilkan <i>list device</i> dengan nama perangkat yang dimasukkan	Sesuai dengan ekspektasi	Lolos
17		Membuat perangkat di <i>master device</i>	1. Pengguna memasukkan data sesuai dengan form yang ada 2. Pengguna menekan tombol buat	<i>Website</i> dapat menyimpan data sesuai dengan data yang dimasukkan	Sesuai dengan ekspektasi	Lolos
18		Mengubah perangkat di <i>master device</i>	1. Pengguna mencari perangkat yang ingin diubah 2. Pengguna memperbaiki data yang ada 3. Pengguna menekan tombol simpan	<i>Website</i> dapat menyimpan data yang baru diubah	Sesuai dengan ekspektasi	Lolos
19		Menghapus perangkat di <i>master device</i>	1. Pengguna mencari data	<i>Website</i> menghapus data	Sesuai dengan ekspektasi	Lolos



			perangkat yang ingin dihapus 2. Pengguna menekan tombol hapus	perangkat yang ingin dihapus		
20		Melihat data hubungan antara <i>sim card</i> dan <i>device</i> di <i>customer premises equipment (CPE)</i>	1. Pengguna menekan halaman CPE	<i>Website</i> menampilkan <i>list CPE</i>	Sesuai dengan ekspektasi	Lolos
21		Mencari <i>Customer Premises Equipment (CPE)</i>	1. Pengguna memasukkan nomor <i>sim card</i> di formulir yang ada 2. Pengguna menekan tombol pencarian	<i>Website</i> berhasil mencari data berdasarkan nomor <i>sim card</i>	Sesuai dengan ekspektasi	Lolos
22	<i>Customer Premises Equipment (CPE)</i>	Mencari <i>Customer Premises Equipment (CPE)</i> menggunakan nama perangkat	1. Pengguna memasukkan nama perangkat di formulir yang ada 2. Pengguna menekan tombol pencarian	<i>Website</i> berhasil mencari data berdasarkan nama <i>device</i>	Sesuai dengan ekspektasi	Lolos
23		Mencari <i>Customer Premises Equipment (CPE)</i> menggunakan email	1. Pengguna memasukkan email di formulir yang ada 2. Pengguna menekan tombol pencarian	<i>Website</i> berhasil mencari data berdasarkan <i>email device</i>	Sesuai dengan ekspektasi	Lolos
24		Membuat <i>Customer Premises Equipment (CPE)</i>	1. Pengguna menekan “ <i>add CPE</i> ” 2. Pengguna mengisi data sesuai <i>simcard</i> dan perangkat yang baru didaftarkan	<i>Website</i> dapat menyimpan data CPE	Sesuai dengan ekspektasi	Lolos

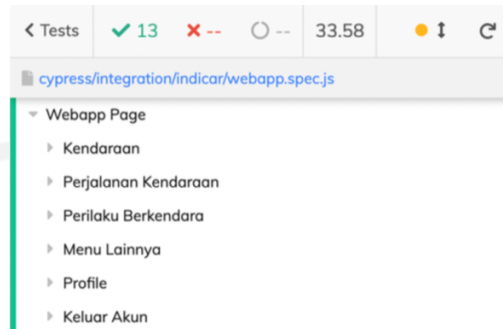
25		Mengubah <i>Customer Premises Equipment (CPE)</i>	1. Pengguna mencari data CPE yang ingin diubah 2. Pengguna menekan tombol ubah dan perbarui data yang ada 3. Pengguna menekan tombol simpan	<i>Website</i> berhasil memperbarui data CPE	Sesuai dengan ekspektasi	Lolos
26		Menghapus <i>Customer Premises Equipment (CPE)</i>	1. Pengguna mencari data CPE yang ingin dihapus 2. Pengguna menekan tombol hapus	<i>Website</i> berhasil menghapus data CPE	Sesuai dengan ekspektasi	Lolos
27		Menampilkan data <i>manufacture</i> di <i>master manufacture</i>	1. Pengguna menekan halaman <i>master manufacture</i>	<i>Website</i> berhasil mengalihkan halaman ke halaman <i>manufacture</i> dan menampilkan data <i>manufacture</i>	Sesuai dengan ekspektasi	Lolos
28	<i>Manufacture</i>	Mencari data <i>manufacture</i> di <i>master manufacture</i>	1. Pengguna memasukkan nama <i>manufacture</i> di form yang ada 2. Pengguna menekan tombol pencarian	<i>Website</i> berhasil mencari data berdasarkan nama <i>manufacture</i>	Sesuai dengan ekspektasi	Lolos
29		Membuat data <i>manufacture</i> di <i>master manufacture</i>	1. Pengguna menekan “ <i>add Manufacture</i> ” 2. Pengguna mengisi data sesuai form <i>manufacture</i> yang ingin didaftarkan	<i>Website</i> dapat menyimpan data <i>manufacture</i>	Sesuai dengan ekspektasi	Lolos
30		Mengubah data <i>manufacture</i> di <i>master manufacture</i>	1. Pengguna mencari data <i>manufacture</i> yang ingin diubah	<i>Website</i> berhasil mengubah data <i>manufacture</i>	Sesuai dengan ekspektasi	Lolos

			2. Pengguna menekan tombol ubah dan perbarui data yang ada 3. Pengguna menekan tombol simpan			
31	Model kendaraan	Menampilkan data <i>model</i> di <i>master model</i>	1. Pengguna menekan halaman <i>master model</i>	<i>Website</i> berhasil mengalihkan halaman ke halaman <i>model</i> dan menampilkan data <i>model</i>	Sesuai dengan ekspektasi	Lolos
32		Mencari data <i>model</i> di <i>master model</i>	1. Pengguna memasukkan nama <i>model</i> di form yang ada 2. Pengguna menekan tombol pencarian	<i>Website</i> berhasil mencari data berdasarkan nama <i>model</i>	Sesuai dengan ekspektasi	Lolos
33		Membuat data <i>model</i> di <i>master model</i>	1. Pengguna menekan “ <i>add model</i> ” 2. Pengguna mengisi data sesuai form <i>model</i> yang ingin didaftarkan	<i>Website</i> dapat menyimpan data <i>model</i>	Sesuai dengan ekspektasi	Lolos
34		Mengubah data <i>model</i> di <i>master model</i>	1. Pengguna mencari data <i>model</i> yang ingin diubah 2. Pengguna menekan tombol ubah dan perbarui data yang ada 3. Pengguna menekan tombol simpan	<i>Website</i> berhasil memperbarui data <i>model</i>	Sesuai dengan ekspektasi	Lolos
35		Menghapus data <i>model</i> di <i>master model</i>	1. Pengguna mencari data <i>model</i> yang ingin dihapus 2. Pengguna menekan tombol hapus	<i>Website</i> berhasil menghapus data <i>model</i>	Sesuai dengan ekspektasi	Lolos

36	<i>Manufacture</i>	Menghapus data <i>manufacture</i> di <i>master manufacture</i>	1. Pengguna mencari data <i>manufacture</i> yang ingin dihapus 2. Pengguna menekan tombol hapus	<i>Website</i> berhasil menghapus data <i>manufacture</i>	Sesuai dengan ekspektasi	Lolos
37	Profil	Menampilkan data <i>profile</i>	1. Pengguna menekan tombol <i>profile</i>	<i>Website</i> menampilkan data <i>profile</i>	Sesuai dengan ekspektasi	Lolos
38		Mengubah data <i>profile</i>	1. Pengguna menekan tombol “ <i>edit profile</i> ” 2. Pengguna mengubah data yang ingin diubah	<i>Website</i> berhasil menyimpan data <i>profile</i> yang diperbarui	Sesuai dengan ekspektasi	Lolos
39		Mengecek data perubahan	1. Pengguna menekan tombol <i>profile</i> 2. Pengguna melihat data yang berhasil diubah	<i>Website</i> berhasil menampilkan data <i>profile</i> yang terbaru	Sesuai dengan ekspektasi	Lolos
40		Mengubah foto <i>profile</i>	1. Pengguna menekan tombol “ <i>change picture</i> ” 2. Pengguna memilih gambar sebagai foto <i>profile</i>	<i>Website</i> berhasil mengganti foto <i>profile</i> yang diinginkan	Sesuai dengan ekspektasi	Lolos
41		Mengubah kata sandi	1. Pengguna menekan tombol “ <i>change password</i> ” 2. Pengguna memasukkan <i>password</i> yang baru	<i>Website</i> berhasil mengganti <i>password</i> akun	Sesuai dengan ekspektasi	Lolos
42		Autentikasi	Keluar dari akun	1. Pengguna menekan tombol “Keluar”	<i>Website</i> berhasil mengeluarkan akun	Sesuai dengan ekspektasi

Pada Tabel 3.3 terdapat 42 pengujian pada website admin IndiCar yang sesuai dengan ekspektasi yang diharapkan. Akan tetapi, pengujian yang penulis cantumkan di sini adalah

pengujian terakhir sebelum *website* admin IndiCar digunakan oleh para pelanggannya. Sama halnya pada Gambar 3.13 pada aplikasi webapp IndiCar, semua pengujian yang penulis cantumkan merupakan pengujian terakhir sebelum digunakan untuk melacak kendaraan dari setiap akun pelanggan.



Gambar 3.13 Hasil test runner Webapp IndiCar

Lain halnya dengan pengujian yang dilakukan pada website webapp IndiCar yang terdiri 6 kategori dengan 13 pengujian. Lebih detail dari pengujian yang dilakukan pada webapp IndiCar pada Tabel 3.4 Selama proses pengujian webapp IndiCar berlangsung membutuhkan waktu 33,58 detik. Detail dari pengujian pada webapp IndiCar dapat dilihat pada Tabel 3.4.

Tabel 3.4 Detail pengujian pada Webapp IndiCar

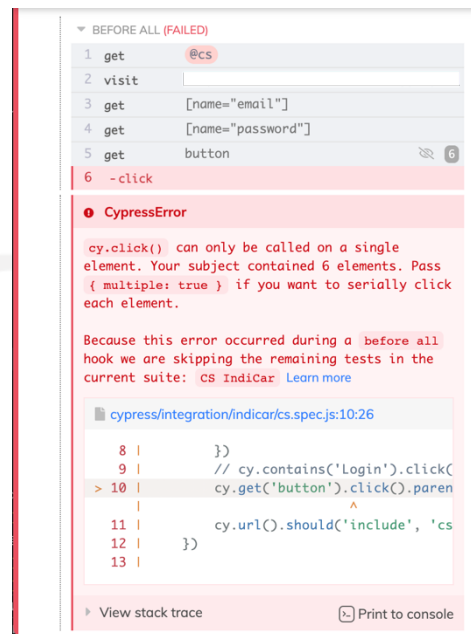
No	Kategori	Skenario Uji	Langkah-langkah	Ekspektasi	Hasil Realita	Status
1	Autentikasi	Login	1. Buka <i>website</i> webapp IndiCar 2. Masukkan email dan <i>password</i> 3. Klik login	Pengguna dapat masuk ke akun	Sesuai dengan ekspektasi	Lolos
2	Kendaraan	Melihat kendaraan	1. Pengguna menekan tombol kendaraan	<i>Website</i> dapat menampilkan data kendaraan dan lokasinya	Sesuai dengan ekspektasi	Lolos
3		Melihat histori perjalanan	1. Pengguna menekan menu "Perjalanan" 2. Pengguna memilih kendaraan yang ingin dilacak 3. Pengguna memilih	<i>Website</i> menampilkan data perjalanan di peta dan dapat menjalankan simulasi histori perjalanan	Sesuai dengan ekspektasi	Lolos

			tanggal yang ingin dilihat riwayatnya			
4	Kebiasaan pengguna	Melihat perilaku berkendara pengguna	1. Pengguna menekan menu “Perilaku Berkendara” 2. Pengguna memilih kendaraan yang ingin dilacak 3. Pengguna memilih tanggal yang ingin dilihat riwayatnya	<i>Website</i> menampilkan data seputar cara berkendara pengguna	Sesuai dengan ekspektasi	Lolos
5		Melihat skor berkendara pengguna	1. Pengguna menekan menu “Menu Lainnya” dan menekan menu “Skor Berkendara” 2. Pengguna memilih kendaraan yang ingin dilacak 3. Pengguna memilih tanggal yang ingin dilihat skor-nya	<i>Website</i> menampilkan <i>score</i> dari cara pengguna berkendara	Sesuai dengan ekspektasi	Lolos
6	Kondisi kendaraan	Melihat kondisi kendaraan pengguna	1. Pengguna menekan menu “Menu Lainnya” dan menekan menu “Kondisi Kendaraan” 2. Pengguna memilih kendaraan yang ingin dilacak 3. Pengguna memilih tanggal yang ingin dilihat skor-nya	<i>Website</i> menampilkan data kondisi kendaraan	Sesuai dengan ekspektasi	Lolos
7	Notifikasi	Menampilkan list notifikasi di menu lainnya	1. Pengguna menekan menu lainnya dan menekan notifikasi	<i>Website</i> menampilkan list notifikasi berdasarkan	Sesuai dengan ekspektasi	Lolos

			2. Pengguna menekan kategori notifikasi pada sisi kiri website	kategori yang ada		
8	Profil	Menampilkan data profil	1. Pengguna menekan tombol <i>profile</i>	Website menampilkan data profil	Sesuai dengan ekspektasi	Lolos
9		Mengubah data profil	1. Pengguna menekan tombol “ <i>edit profile</i> ” 2. Pengguna mengubah data yang ingin diubah	Website berhasil menyimpan data profil yang diperbarui	Sesuai dengan ekspektasi	Lolos
10		Mengecek data perubahan	1. Pengguna menekan tombol <i>profile</i> 2. Pengguna melihat data yang berhasil diubah	Website berhasil menampilkan data profil yang terbaru	Sesuai dengan ekspektasi	Lolos
11		Mengubah foto profil	1. Pengguna menekan tombol “ <i>change picture</i> ” 2. Pengguna memilih gambar sebagai foto <i>profile</i>	Website berhasil mengganti foto profil yang diinginkan	Sesuai dengan ekspektasi	Lolos
12		Mengubah kata sandi	1. Pengguna menekan tombol “ <i>change password</i> ” 2. Pengguna memasukkan <i>password</i> yang baru	Website berhasil mengganti <i>password</i> akun	Sesuai dengan ekspektasi	Lolos
13		Autentikasi	Keluar dari akun	1. Pengguna menekan tombol “Keluar”	Website berhasil mengeluarkan akun	Sesuai dengan ekspektasi

Sedangkan status kesalahan pada Cypress biasanya bernama *CypressError*. *CypressError* adalah sebuah kesalahan yang ada di Cypress apabila kode yang telah dibuat tidak dapat dijalankan atau hasil yang dimunculkan tidak sesuai dengan kode yang ditulis. Saat

terjadi kesalahan yang tidak sesuai dengan data kode yang telah dibuat akan ditampilkan seperti Gambar 3.14.



```

1 get @cs
2 visit
3 get [name="email"]
4 get [name="password"]
5 get button
6 -click

CypressError

cy.click() can only be called on a single
element. Your subject contained 6 elements. Pass
{ multiple: true } if you want to serially click
each element.

Because this error occurred during a before all
hook we are skipping the remaining tests in the
current suite: CS IndiCar Learn more

cypress/integration/indicar/cs.spec.js:10:26

8 |     })
9 |     // cy.contains('Login').click()
> 10 |     cy.get('button').click().paren
    |                               ^
11 |     cy.url().should('include', 'cs
12 |     })
13 |

View stack trace Print to console

```

Gambar 3.14 Contoh tampilan kesalahan di *test-runner* Cypress

Pada Gambar 3.14 merupakan salah satu contoh bahwa kode dari Cypress yang telah dibuat tidak sesuai dengan website yang akan diuji. Pada gambar tersebut menyebutkan bahwa *parent* yang ada di *button* tidak dijelaskan. Lain halnya apabila *website* tidak dapat dapat memuat data yang diinginkan maka akan ditampilkan di terminal seperti pada Gambar 3.15.



```

-----
Command timed out after 30000 milliseconds:
Timed out
-----

```

Gambar 3.15 Tampilan di terminal Cypress

### 3.3.6. Hasil Eksekusi

Berdasarkan eksekusi yang telah dilakukan pada website IndiCar admin dan webapp, semua pengujian berhasil dilakukan tanpa ada kesalahan. Hal ini dapat terjadi karena tampilan dari IndiCar tidak berubah. Apabila terjadi perubahan pada tampilan IndiCar, maka akan terjadi kesalahan yang menyebabkan tampilan kode Cypress harus dibuat ulang. Apabila tidak terdapat perbedaan dari segi tampilan maka dapat dilaporkan ke *developer* ketika ditemukannya kesalahan ataupun telah sesuai dengan hasil yang diharapkan. Kesalahan tersebut akan diperbaiki saat proses *sprint* berlangsung ataupun di *sprint* selanjutnya.



## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1. Relevansi Akademik

Berdasarkan teori yang telah dikaji sebelumnya, maka penulis menemukan suatu variasi nilai dan kesenjangan antara teori dengan fakta empiris di lapangan, yaitu pada pengujian *end-to-end*, Cypress, *scrum*, dan *sprint*.

##### 4.1.1. Pengujian *End-to-End*

Secara teori pengujian *end-to-end* merupakan sebuah pengujian yang dilakukan pada website yang dibuat ataupun alat yang dipasangkan di lapangan (Huoponen, 2020). Selama magang berlangsung yang dapat dilakukan hanya menguji website IndiCar dan tidak dapat langsung menguji alat IndiCar yang dipasangkan pada kendaraan. Agar dapat menjalankan pengujian *end-to-end* secara sempurna, dibutuhkan kerja sama dengan para pihak yang ada di lapangan untuk membantu memastikan bahwa data yang ada di website sesuai dengan data yang ada di lapangan. Berbagai kesulitan tidak dapat dihindari dalam pelaksanaan pengujian.

Tidak hanya itu, secara teori penggunaan perkakas pengujian *end-to-end* bernama Cypress sangat berguna dalam pengujian ini karena memiliki fitur yang sangat lengkap (Alajävi, 2022). Sayangnya, selama magang berlangsung penulisan kode sangat sulit dilakukan karena Cypress sangat tergantung pada suatu elemen tertentu dalam melakukan sesuatu. Apabila suatu elemen diganti, maka penulisan kode Cypress perlu ditulis ulang dan tidak dapat selesai yang bersamaan dengan selesainya proses pembuatan tampilan pada aplikasi IndiCar. Agar hal ini tidak terjadi, penulisan Cypress sebaiknya dilakukan saat pembuatan tampilan telah selesai dilakukan.

##### 4.1.2. Pengujian dengan Cypress

Cypress adalah sebuah perkakas pengujian *end-to-end* gratis dan memiliki tampilan visual dalam melakukan pengujian (Taky, 2021). Dengan adanya hal ini, penulis dapat mengamati proses pengujian melalui *dashboard* yang iteratif tanpa adanya batasan pengujian. Selain itu, penulis juga dapat menghemat waktu, tenaga, serta biaya untuk melakukan pengujian yang berulang.

Selain itu, Cypress memiliki fitur yang sangat banyak seperti mengunggah foto ataupun file ke dalam website IndiCar. Hal ini sangat bermanfaat bagi penulis karena dalam pengujian

IndiCar terdapat beberapa pengujian yang memerlukan mengunggah gambar. Sayangnya, fitur yang ada di Cypress tersebut tergantung pada sebuah elemen tertentu sehingga penulisan ulang kode Cypress saat terjadi perubahan atau penambahan dari segi tampilan. Hal itu yang dirasakan oleh penulis ketika membuat kode Cypress saat proses pengembangan aplikasi. Selain penulis, hal itu juga dirasakan oleh Alajävi dalam melakukan pengujian ketika pengujian terhenti karena kode Cypress yang dibuat tidak menemukan elemen di website yang sedang diuji (Alajävi, 2022). Akan tetapi, penulis tidak khawatir akan hal tersebut karena saat terjadi kesalahan yang penyebabnya tidak diketahui penulis melakukan pencarian di Google dan permasalahan tersebut dapat terselesaikan. Hal ini dapat terjadi karena Cypress memiliki komunitas yang luas sehingga dapat berbagi pengalaman serta solusi yang mereka ketahui.

#### 4.1.3. Perbandingan Cypress

Terdapat perangkat pengujian *end-to-end* selain Cypress yang memiliki ciri khasnya sendiri. Setiap perangkat memiliki kelebihan serta kekurangan yang mana akan dibandingkan dengan Cypress. Perangkat-perangkat tersebut adalah Selenium webdriver, Katalon Studio, dan testIM.

Selenium webdriver adalah sebuah perangkat sejenis Cypress yang berguna untuk melakukan pengujian end-to-end secara otomatis dan dapat berjalan dengan koneksi yang kurang bagus. Selenium webdriver mendukung banyak browser serta dapat digunakan untuk aplikasi Ajax. Selenium webdriver juga memiliki banyak perintah di dalamnya serta memiliki komunitas yang luas yang dapat membantu apabila terjadi kesalahan. Akan tetapi, Selenium webdriver tidak memiliki *dashboard* yang dapat memvisualisasikan pengujian yang sedang dilakukan.

Perangkat selanjutnya bernama Katalon Studio, Katalon Studio adalah sebuah perangkat pengujian end-to-end otomatis yang menggunakan metode *scripting* berbahasa Java yang dapat digunakan untuk pengujian web, desktop, hingga seluler. Selain itu, Katalon Studio juga sudah memiliki komunitas yang besar sehingga mudah untuk dicari solusi apabila terjadi error. Akan tetapi, penggunaan Katalon Studio sulit untuk dipelajari karena memiliki kode perintah yang cukup panjang dan memerlukan kualitas internet yang bagus saat melakukan pengujian.

Perangkat terakhir yang ingin penulis bandingkan dengan Cypress adalah testIM, testIM memiliki fungsi yang sama seperti Cypress. Berbeda dengan Cypress, metode yang digunakan pada perangkat testIM adalah metode *record&play* yang mana penggunaannya harus merekam hal yang harus dilakukan menggunakan BlazeMeter dan hasilnya dapat dijalankan melalui

*dashboard* interaktif pada website testIM. Dalam hal ini, akan berakibat pada fitur yang ada pada perkakas testIM sangat terbatas. Lalu, apabila penggunaan testIM melebihi 1000 kali, maka akan dikenakan biaya berlangganan. Selain itu, testIM masih tergolong baru sehingga belum mempunyai komunitas yang besar.

Tabel 4.1 Perbandingan Cypress dengan Perkakas Serupa

Perbandingan	Perkakas Pengujian Otomatis			
	Cypress	Selenium webdriver	Katalon Studio	testIM
Dashboard yang iteratif	√	X	√	√
Mudah digunakan	√	√	X	√
Gratis	√	√	√	X
Komunitas yang luas	√	√	√	X
Fitur yang lengkap	√	√	√	X
Dapat digunakan dengan kualitas sinya yang kurang bagus	√	√	X	√

#### 4.1.4. Proses Migrasi Server

Secara teori, sebelum melakukan proses migrasi perlu adanya pemilihan server yang tepat dan setelah selesai melakukannya akan menyusun ulang topologi di server sebelumnya (Hartono dkk., 2016). Sedangkan di Telkom Indonesia khususnya tim IndiCar tidak hanya memilih server yang tepat melainkan juga berkoordinasi dengan tim lainnya untuk membantu proses migrasi seperti tim techops (yang mengurus DevOps) dan tim DbA (yang mengurus database). Setiap tim nantinya akan membuat daftar yang akan dilakukan saat proses migrasi berlangsung. Tidak hanya itu, setiap tim harus latihan untuk menghasilkan waktu yang singkat dalam mengerjakan tugas migrasi yang didapatkan.

Saat proses migrasi berlangsung, semua tim membuka Google Sheets yang telah dibuat bersama dan semuanya memperhatikan tahapan apa yang sedang dikerjakan. Setiap daftar

tugas selesai dikerjakan akan dicatat waktunya. Setelah semua tugas berhasil diselesaikan, IndiCar tidak menyusun ulang topologi di server sebelumnya.

#### 4.1.5. *Scrum*

Dalam penerapan *scrum* terdapat kesulitan di mana *scrum* harus mempunyai dokumentasi yang lengkap untuk memudahkan dalam menyelenggarakan demo produk pada pelanggan. Hal ini berkaitan transparansi antara anggota tim dalam memudahkan anggota lain untuk melihat fitur yang dikembangkan (Alshammari, 2022). IndiCar menggunakan metode *scrum* dalam proses pembuatan aplikasi, namun dalam penerapan *scrum* di tim IndiCar seringkali dokumentasi tidaklah lengkap. Hal ini terjadi karena untuk mendokumentasikan suatu aplikasi membutuhkan waktu, sedangkan aplikasi IndiCar ini sangat dibutuhkan dan diharapkan dapat langsung menggantikan sistem lama yang ada di Telkom Indonesia. Dengan tidak adanya dokumentasi fitur yang dikembangkan, maka penulis sebagai penguji aplikasi tidak dapat menemukan kesalahan pada fitur yang dibuat dan membawa akibat dapat ditemukannya kesalahan tersebut oleh pengguna pada saat menggunakan aplikasi IndiCar. Dengan demikian dokumentasi fitur sangat dibutuhkan dalam pengujian aplikasi.

Dalam penerapan *scrum* juga harus menggunakan *sprint* untuk memberikan batasan waktu terhadap fitur baru yang diinginkan oleh pelanggan (Jondya dkk., 2022). Di IndiCar proses *sprint* dilakukan secara dua minggu yang terbagi menjadi 4 bagian, yaitu *sprint planning*, *daily sprint*, *sprint review*, dan *sprint retrospective*. Namun proses *sprint* yang ada di tim IndiCar terkadang tidak dilakukan secara benar karena developer juga mengerjakan website selain IndiCar yang menyebabkan kurang fokus ke pengerjaan website IndiCar. Dengan adanya hal ini, maka fitur yang dibuat oleh developer tidak dapat terselesaikan dengan cepat yang berakibat pelanggan IndiCar pindah ke sistem sejenis. Dengan demikian *sprint* pada proses *scrum* sangat diperlukan untuk memastikan bahwa pelanggan IndiCar tidak pindah ke sistem lainnya.

## 4.2. Pembelajaran Magang

Magang yang telah dilaksanakan membawa manfaat bagi penulis, yaitu mengerti cara menguji aplikasi menggunakan metode *blackbox testing* khususnya *end-to-end testing*, mengetahui tahapan migrasi server, dan mengerti menggunakan perkakas *end-to-end* seperti Cypress. Tidak hanya itu, adapun manfaat lain yang di dapatkan seperti gratis training online

di website Pijar yang di dalamnya terdapat kursus mahir *scrum master*, membuat cv yang dicari, dan masih banyak lagi.

Selain itu, Telkom Indonesia juga menerapkan metode-metode pengembangan seperti scrum dalam proses pembuatan aplikasi. Dalam kesempatan ini, penulis menjadi lebih mengetahui tentang scrum secara mendalam, seperti tahapan pelaksanaan metode scrum, sistem kerja scrum dan masih banyak lagi yang tidak digunakan saat pembelajaran di kampus. Tidak hanya itu, penulis juga mendapatkan materi yang mendalam tentang *end-to-end testing*, *black box testing* dan ilmu lainnya tidak penulis dapatkan secara mendalam di lingkungan kampus apabila tidak mengambil mata kuliah yang berkaitan. Dengan ilmu-ilmu yang didapatkan selama magang, penulis merasa sangat beruntung karena telah merasakan dunia kerja yang mana tidak didapatkan selama berkuliah.

Akan tetapi selama penulis melaksanakan magang terdapat beberapa kendala, seperti sulitnya melakukan pengujian *end-to-end* pada *website* dengan benar. Hal itu disebabkan karena tim IndiCar tidak mempunyai seorang *tester* maka terdapat kendala dalam hal wawasan serta pengetahuan tentang apa yang perlu dilakukan. Akan tetapi, untuk menutupi semua kekurangan itu supervisor memberikan arahan kepada penulis untuk mempermudah dalam melakukan pengujian serta memberikan referensi apa yang akan dikerjakan.

Walaupun penulis telah diberi petunjuk oleh tim developer, tetap saja terdapat kendala dalam mencari cara yang tepat saat untuk mengerjakan tugas yang diberikan. Hal itu dikarenakan para tim developer sibuk dengan pekerjaannya masing-masing dan para pemegang lainnya memiliki latar belakang yang berbeda serta pekerjaan yang berbeda. Tidak hanya itu, penulis juga kesulitan dalam hal membagi waktu karena penulis dituntut untuk bisa memahami banyak hal. Dengan mempelajari banyak hal tanpa adanya bimbingan yang lebih jelas. Oleh karena itu, penulis hanya mendapatkan ilmu luarnya saja tidak dapat memperdalam ilmu-ilmu yang dipelajari selama magang di Telkom Indonesia.

Hal ini berdampak pada kinerja yang dihasilkan, seperti kurangnya informasi yang didapatkan di rapat harian dan juga lambatnya mengerjakan tugas magang karena bentrok dengan tugas yang diberikan di kampus. Untuk mengatasi hal ini, penulis sudah belajar cara membuat jadwal apa yang harus dikerjakan dan juga jam berapa akan selesai dengan menggunakan aplikasi foca untuk memberikan ruang fokus untuk mengerjakan suatu tugas.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Pengujian otomatis menggunakan metode *end-to-end testing* dapat dipermudah menggunakan Cypress karena dapat mempercepat proses pengujian. Adapun tahapan yang dilakukan selama pembuatan pengujian *end-to-end* menggunakan Cypress seperti pembuatan skenario pengujian, pemilihan perkakas pengujian, validasi scenario pengujian, pembuatan kode Cypress, mengeksekusi kode pengujian, dan memberikan hasil kepada *developer*. Tidak hanya itu, Cypress memiliki *dashboard* yang iteratif untuk melihat kesalahan yang ada pada saat pengujian. *Dashboard* tersebut mendukung beberapa browser untuk menjalankan pengujian.

Akan tetapi, dalam pengujian ini perlu adanya penyesuaian kembali saat ada pengembangan aplikasi. Hal ini membutuhkan waktu yang cukup banyak apabila terjadi suatu pengembangan yang berkaitan dengan tampilan.

#### **5.2. Saran**

Terdapat saran untuk penelitian yang selanjutnya yaitu melakukan pengukuran kecepatan Cypress dalam mengeksekusi *end-to-end testing*. Pengujian ini dapat dipercepat dengan adanya penggunaan algoritma yang lebih tepat. Tidak hanya itu, sebaiknya menggunakan versi Cypress terbaru dalam pembuatan *script testing* karena adanya fitur yang lebih lengkap dibandingkan versi yang penulis gunakan. Kemudian, perlu adanya penelitian yang membandingkan perkakas pengujian otomatis dalam pengujian *end-to-end* atau metode lainnya secara mendalam.



## DAFTAR PUSTAKA

- Alajävi, O. (2022). Ohjelmiston E2E-testaus [Bachelor's thesis / ThesisAMK]. Dalam *Satakunta University of Applied Sciences*. Satakunta University of Applied Sciences.
- Alam, E. N. (2020). IoT in Agriculture Industry. *Jurnal Sistem Cerdas*, 03(01), 36–42. <https://apic.id/jurnal/index.php/jsc/article/view/57/42>
- Alshammari, F. H. (2022). Analytical Evaluation of SOA and SCRUM Business Process Management Approaches for IoT-Based Services Development. *Scientific Programming*, 2022. <https://doi.org/10.1155/2022/3556809>
- Anggris, M. F., Tri Ananta, M., & Muslimah Az-Zahra, H. (2018). Rancang Bangun Aplikasi Augmented Reality Pengelolaan Rambu-Rambu Lalu Lintas Menggunakan Global Positioning System (GPS) pada Android. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(8), 2892–2901. <http://j-ptiik.ub.ac.id>
- Arcuri, A. (2019). RESTful API Automated Test Case Generation with EvoMaster. *ACM Transactions on Software Engineering and Methodology*, 28(1), 1–37. <https://doi.org/10.1145/3293455>
- Badal, L., & Grünler, D. (2021). *Automated end-to-end testing: Useful practice or frustrating time sink?*
- Badan Pusat Statistik. (2021). *Statistik Kriminal 2021*. <https://www.bps.go.id/publication/download.html?nrbvfeve=OGQxYmM4NGQyMDU1ZTk5ZmVlZDM5OTg2&xzmn=aHR0cHM6Ly93d3cuYnBzLmdvLmlkL3B1YmtpY2F0aW9uLzlwMjEvMTIvMTUvOGQxYmM4NGQyMDU1ZTk5ZmVlZDM5OTg2L3N0YXRpc3RpaylrcmltaW5hbC0yMDIxLmh0bWw%3D&twoadfnoarfeauf=MjAyMi0xMC0zMSAwMzo0NTozOQ%3D%3D>
- DDB Telkom. (2022). *Pisah Sambut PMMB FHCI BUMN Juni 2022*. <https://www.youtube.com/c/InternshipDDBTelkom>
- Fadli, A. (2020). Mengenal COVID-19 dan Cegah Penyebarannya Dengan “Peduli Lindungi” Aplikasi Berbasis Android. Dalam *ResearchGate*. [https://www.researchgate.net/profile/Ari-Fadli/publication/340790225\\_MENGENAL\\_COVID-19\\_DAN\\_CEGAH\\_PENYEBARANNYA\\_DENGAN\\_PEDULI\\_LINDUNGI\\_APLIKASI\\_BERBASIS\\_ANDORID/links/5e9e6252a6fdcca7892bd9f9/MENGENAL-COVID-](https://www.researchgate.net/profile/Ari-Fadli/publication/340790225_MENGENAL_COVID-19_DAN_CEGAH_PENYEBARANNYA_DENGAN_PEDULI_LINDUNGI_APLIKASI_BERBASIS_ANDORID/links/5e9e6252a6fdcca7892bd9f9/MENGENAL-COVID-)

19-DAN-CEGAH-PENYEBARANNYA-DENGAN-PEDULI-LINDUNGI-APLIKASI-BERBASIS-ANDORID.pdf

- Firman Ashari, I., Aryani, A. J., & Ardhi, A. M. (2022). Design and Build Inventory Management Information System Using The Scrum Method. *JSiI (Jurnal Sistem Informasi)*, 9(1), 27–35.
- Govil, N., & Sharma, A. (2022). Validation of agile methodology as ideal software development process using Fuzzy-TOPSIS method. *Advances in Engineering Software*, 168, 103125. <https://doi.org/10.1016/j.advengsoft.2022.103125>
- Gowda, M., Manweiler, J., Dhekne, A., Choudhury, R. R., & Weisz, J. D. (2016). Tracking drone orientation with multiple GPS receivers. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 0(1), 280–293. <https://doi.org/10.1145/2973750.2973768>
- Hartono, N., Utami, E., & Amborowati, A. (2016). Hartono, Migrasi dan Optimalisasi Database Sistem Informasi Manajemen Universitas Cokroaminoto Palopo 255 Migrasi dan Optimalisasi Database Sistem Informasi Manajemen Universitas Cokroaminoto Palopo. *Jurnal Buana Informatika*, 7(4), 255–264.
- Hlaing, N. N. S., Naing, M., & Naing, S. S. (2019). GPS and GSM Based Vehicle Tracking System. *Published in International Journal of Trend in Scientific Research and Development*, 3(4), 271–275. <http://creativecommons.org/licenses/by/4.0>
- Huoponen, J. (2020). *Web-Sovelluksen Päästä Päähän -Testaaminen Cypress-Testaustyökalulla*.
- Jondya, A. G., Anwar, H. I., Hermawan, M. R., & Deru, T. A. (2022). Developing “EMBER”: A Fantasy Puzzle Adventure Game with An Approach to Agile by Applying Game-Scrum Methodology. *Engineering, Mathematics and Computer Science (EMACS) Journal*, 4(2), 39–43. <https://doi.org/10.21512/emacsjournal.v4i2.8428>
- K. Moorthi, P., Shrivastava, P., & Krishnan, S. (2018). Transesterification of Moringa Oleifera Seed Oil by Sodium Silicate Catalyst Using Different Co-Solvents. *International Journal of Engineering & Technology*, 7(3.36), 1. <https://doi.org/10.14419/ijet.v7i3.36.29068>
- Ketaren, A. B., & Barus, E. S. (2020). Server Statistika Jumlah Jemaat GBKP Klasis Menara (Medan-Namorambe) Berbasis Web. *PUBLIKASI ILMIAH TEKNOLOGI INFORMASI NEUMANN (PITIN)*, 5(2), 52–58.



- Mahomed, F., Swanepoel, D. W., Eikelboom, R. H., & Soer, M. (2021). Validity of automated threshold audiometry: a systematic review and meta-analysis. *Ear & Hearing*. <https://doi.org/10.1097/AUD.0b013e3182944bdf>
- Muslihah, I., Akbar, N., & Rais, R. (2021). Sistem Informasi Akuntansi Keuangan Sekolah Berbasis Client Server. *JDMSI*, 2(2), 45–55.
- Nguyen, H. (2022). *Single-Page Application And Front-End Testing Methods-Built with React and React Router, Tested with Jest and Cypress*. Turku University of Applied Sciences.
- Rahmadhani, V., Arum, W., Bhayangkara, U., & Raya, J. (2022). *Literature Review Internet Of Think (IOT): Sensor, Konektifitas Dan QR Code*. 3(2). <https://doi.org/10.38035/jmpis.v3i2>
- Rix, A. J., Abraham, C. J., & Booyesen, M. J. (2022). Why taxi tracking trumps tracking passengers with apps in planning for the electrification of Africa's paratransit. *IScience*, 25(9), 104943. <https://doi.org/10.1016/j.isci.2022.104943>
- Snadhika Jaya, T. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(02).
- Sukmawati, R., & Priyadi, Y. (2019). Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, 3(2), 104. <https://doi.org/10.29407/intensif.v3i2.12697>
- Syafriadi, Kurniadi, W., & Nirsal, dan. (2021). Optimalisasi Layanan Infratructure as a Service Dengan Proxmox pada Server Universitas Cokroaminoto Palopo. *Optimalisasi Layanan Infratructure as a Service Dengan Proxmox Pada Server Universitas Cokroaminoto Palopo*, 284–287.
- Taky, M. T. (2021). Automated Testing With Cypress. Dalam *VAASAN AMMATTIKORKEAKOULU UNIVERSITY OF APPLIED SCIENCES Information Technology*. VAASAN AMMATTIKORKEAKOULU UNIVERSITY OF APPLIED SCIENCES Information Technology.
- Telkom Indonesia. (t.t.). *IndiCar*.
- Telkom Indonesia. (2020). *Profil dan Riwayat Singkat Telkom Indonesia*. [https://www.telkom.co.id/sites/about-telkom/id\\_ID/page/profil-dan-riwayat-singkat-22](https://www.telkom.co.id/sites/about-telkom/id_ID/page/profil-dan-riwayat-singkat-22)
- Telkom Indonesia. (2022). *Penghargaan Telkom Indonesia*. [https://www.telkom.co.id/sites/about-telkom/id\\_ID/page/penghargaan-649](https://www.telkom.co.id/sites/about-telkom/id_ID/page/penghargaan-649)

- Tessarolo, F., Petsani, D., Conotter, V., Nollo, G., Conti, G., Nikolaidou, M., Onorati, G., Bamidis, P. D., & Konstantinidis, E. I. (2022). Developing ambient assisted living technologies exploiting potential of user-centred co-creation and agile methodology: the CAPTAIN project experience. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-021-03649-0>
- Villamil, S., Hernández, C., & Tarazona, G. (2020). An overview of internet of things. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(5), 2320–2327. <https://doi.org/10.12928/TELKOMNIKA.v18i5.15911>
- Wahyuningtyas, S. Y. (2019). Self-regulation of online platform and competition policy challenges: A case study on Go-Jek. *Competition and Regulation in Network Industries*, 20(1), 33–53. <https://doi.org/10.1177/1783591719834864>
- Wijayanto, D., Firdonsyah, A., Dharma Adhinata, F., & Jayadi, A. (2021). Rancang Bangun Private Server Menggunakan Platform Proxmox dengan Studi Kasus: PT.MKNT. *Jurnal ICTEE*, 2(2), 41–49. [www.example.com](http://www.example.com)

LAMPIRAN

