



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Kedudukan dalam perusahaan selama menjalani proses kerja magang adalah sebagai *Frontend Engineer Intern* pada tim *Utilities* di divisi *Tech Platform* dengan mentor yaitu Bapak Iuean Ignatius yang memiliki jabatan sebagai selaku *Mid Frontend Engineer*. Selama menjalani proses kerja magang, seluruh tugas diberikan oleh mentor dan melapor kembali ke mentor apabila ada kendala atau tugas sudah selesai.

Didalam tim *Utilities*, terdapat beberapa *job desk* berbeda dan saling berkaitan untuk mengembangkan fitur *Utilities* pada website tiket.com, yaitu:

- *Technical Lead* bertugas sebagai pemimpin tim dan memberikan arahan terhadap tugas-tugas yang sedang dikerjakan oleh setiap anggota tim.
- *Scrum Master* bertugas mengatur *development life cycle* dalam tim.
- *Product Manager* bertugas mengatur tugas-tugas yang akan dikerjakan oleh tim dalam suatu *sprint* dan membuat prioritas tentang tugas-tugas yang harus segera diselesaikan. *Product Manager* juga bertugas menjembatani antara tim *developer* dan tim *business* maupun tim *design UI/UX*.
- *Quality Assurance* bertugas menguji program yang telah dikembangkan untuk menemukan kesalahan atau bug pada program agar program dapat berjalan tanpa masalah berarti.

- *Backend Engineer* bertugas mengelola data pada basis data, membuat API ataupun *endpoint* untuk diakses oleh *Frontend Engineer*.
- *Frontend Engineer* bertugas mengolah data dari restAPI yang telah dibuat oleh *Backend Engineer* untuk mengembangkan fitur *Utilities* seperti pada halaman utama, promo, maupun menambahkan fitur baru pada halaman website sesuai dengan rancangan antarmuka yang telah dibuat dan juga melakukan unit testing sebelum di serahkan ke bagian *Quality Assurance*.

Manajemen pengembangan piranti lunak atau yang biasa disebut SDLC merupakan manajemen yang digunakan oleh PT Global Tiket Network adalah *agile scrum*. *Scrum* adalah sebuah framework atau kerangka kerja yang bisa digunakan untuk proyek yang memiliki skala besar seperti PT Global Tiket Network. Dalam metode *scrum*, seorang *Product Manager* akan melakukan *listing* terhadap seluruh tugas-tugas yang akan diberikan pada tim *Utilities* dalam sebuah *backlog*. Setiap 2 minggu sekali akan dilakukan *Sprint Planning*, yaitu sebuah rapat untuk menentukan tugas-tugas apa yang akan diambil oleh *developer* dari *backlog* yang telah dibuat sebelumnya oleh *Product Manager*. Setiap tugas memiliki bobot nilai tertentu yang akan digunakan untuk mengevaluasi kerja tim.

Rapat koordinasi lainnya yang dilakukan oleh tim *Utilities* adalah *weekly sync-up* dan *Standup Meeting*. *Weekly sync-up* merupakan rapat mingguan yang dilakukan untuk mengetahui progress maupun kendala apa saja yang dialami oleh tim dalam menjalani tugas dari *sprint planning* sebelumnya. Untuk tim *Utilities*, *weekly sync-up* diadakan rutin setiap hari selasa. Pada *weekly sync-up* juga dilakukan diskusi tentang bagaimana performa tim sejauh ini dan apa saja yang perlu di-*improve* kedepannya.

*Standup Meeting* merupakan rapat harian yang dilakukan oleh seluruh anggota tim *Utilities*. *Standup Meeting* pada tim *Utilities* diadakan setiap pukul 10:30 WIB. Rapat ini diadakan untuk membahas tentang apa saja yang telah dilakukan oleh setiap anggota tim pada hari sebelumnya dan apa yang akan dilakukan hari itu. Tujuan diadakannya rapat harian seperti ini adalah untuk melihat progres harian anggota tim dan kendala apa yang ditemukan pada kemarin agar bisa diselesaikan hari itu.

Dalam menyelesaikan tugas setiap anggota tim menggunakan *git workflow* saat menyelesaikan tugas-tugas yang telah diberikan saat *sprint planning*. Saat memulai tugas baru, anggota tim akan membuat *branch* baru dari *repository* yang dikerjakan. Setelah tugas selesai, anggota tim melakukan *push* ke *branch* baru yang telah dibuat sebelumnya. Selanjutnya, dilakukan *pull request* untuk menggabungkan *branch* baru tersebut dengan *branch development*. Lalu *senior engineer* dan *technical lead* akan melakukan *review* terhadap tugas tersebut. Apabila disetujui, maka tugas baru tersebut selesai dan *branch* tersebut akan digabungkan dengan *branch development*.

### **3.2 Tugas yang dilakukan**

Selama menjalani proses kerja magang di PT Global Tiket Network sebagai *Frontend Engineer Intern*, tugas-tugas yang diberikan adalah tugas-tugas yang berhubungan dengan ranah tim *Utilities*, yaitu bagian halaman utama, halaman promo. Tugas-tugas tersebut diberikan melalui *Sprint Planning* ataupun diberikan langsung oleh mentor atau supervisi. Bagian *Frontend* pada tim *Utilities* menggunakan bahasa pemrograman Javascript dengan framework *reactJs* dan *Es6*

atau ECMAScript. Untuk mengambil data pada database *frontend* menggunakan GraphQL yang berfungsi sebagai jembatan antara *frontend* dan *backend*. Setiap tugas yang telah diselesaikan akan dilakukan pengecekan atau *code review* melalui github. Tugas-tugas yang dilakukan yaitu melakukan pengujian unit testing terhadap detail perancangan program agar program dapat dijalankan sesuai dengan struktur control dari desain program yang telah dibuat mencegah terjadinya celah bug sebelum di serahkan kepada tim QA (Quality Assurance) dan juga melakukan tugas untuk mengubah *footer* dan *header* pada halaman web tiket.com.

Selain mempelajari *reactJs*, ECMAScript versi 6, GraphQL juga mempelajari library dari *reactJs* yaitu Jest. Jest ini yang akan digunakan saat melakukan testing pada software dan ini juga mendukung ES6. Dalam *testing software* salah satunya yaitu dengan menggunakan *White Box*. *White Box Testing* merupakan testing yang dilakukan terhadap kondisi internal dari software tersebut secara langsung. *White Box Testing* ini sangat berkaitan dengan *code coverage*. *Code Coverage* adalah istilah pengukuran untuk menunjukkan seberapa lengkap suatu *source code* yang telah dicakup dalam test yang telah dilakukan. Dalam Code Coverage yang di gunakan ada 4 bagian yaitu, Statement Coverage, Branch Coverage, Function Coverage dan Lines.

### **3.3 Uraian Pelaksanaan Kerja Magang**

#### **3.3.1 Proses Pelaksanaan**

Dalam menjalani proses kerja magang, realisasi kerja magang dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Realisasi Kerja Magang

Minggu ke-	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> <li>• Mempelajari <i>ES6</i> dan ReactJS</li> <li>• Mempelajari struktur pemograman</li> </ul>
2	<ul style="list-style-type: none"> <li>• Mengerjakan Unit Testing pada Landing Page</li> </ul>
3	<ul style="list-style-type: none"> <li>• Mengerjakan Unit Testing pada Landing Page</li> </ul>
4	<ul style="list-style-type: none"> <li>• Mengerjakan Unit Testing pada Landing Page</li> </ul>
5	<ul style="list-style-type: none"> <li>• Mengerjakan Unit Testing pada Landing Page</li> </ul>
6	<ul style="list-style-type: none"> <li>• Mengerjakan Unit Testing pada Landing Page</li> </ul>
7	<ul style="list-style-type: none"> <li>• Mengejaakan Header dan Footer pada Landing Page Desktop</li> </ul>
8	<ul style="list-style-type: none"> <li>• Mengejaakan Header dan Footer pada Landing Page Desktop</li> </ul>
9	<ul style="list-style-type: none"> <li>• Mengejaakan Header dan Footer pada Info Pages</li> </ul>
10	<ul style="list-style-type: none"> <li>• Mengejaakan Header dan Footer pada Info Pages</li> </ul>

### 3.3.1.1 Mempelajari *ES6* dan ReactJS.

Pada awal proses kerja magang, tugas yang pertama diberikan adalah untuk mempelajari bahasa pemrograman serta *framework* dan *tools* lain yang digunakan oleh tim Utilities. Sistem operasi yang digunakan adalah Ubuntu dengan versi 18.04 LTS. Ubuntu digunakan karena memiliki *environment* yang mudah digunakan oleh *developer* untuk menggunakan *threading* dan *caching*. Bahasa pemrograman yang digunakan adalah Proses pembelajaran dilakukan dengan membaca dokumentasi serta menonton video tutorial yang menjelaskan tentang topik-topik tersebut. Setelah mempelajari *framework* reactJS dan ECMAScript 6, dilanjutkan dengan mendalami cara kerja ECMAScript 6. ECMAScript 6 atau ES6 adalah bahasa *scripting* standarisasi (JavaScript) yang

dibuat oleh *European Computer Manufacturers Association* (ECMA). Dalam ES6 ini membutuhkan *tools* untuk menangani masalah *browser* yang belum mendukung penggunaan ES6 yaitu babel dan webpack.

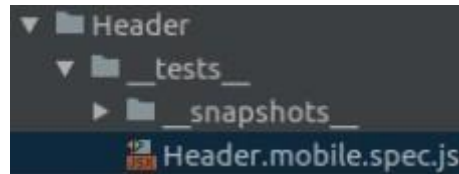
Tidak hanya reactJs dan ES6 tetapi juga mempelajari Jest. Jest adalah pustaka pengujian JavaScript terbuka dari Facebook. Jest ini sederhana, sangat cepat dalam melakukan testing dan juga cukup mudah diinstal dengan menggunakan npm atau yarn (Gigi Sayfan , 2018). Selain itu Jest mempunyai fungsi mock yang sederhana. Mock atau mocking adalah bagian yang sangat penting dari pengujian unit pada suatu sistem. Ada beberapa mock yang ada pada Jest seperti:

1. `jest.fn()` ialah mock yang berfungsi untuk memberikan nilai kembali untuk panggilan yang berturut-turut dan mencatat berapa kali itu disebut dan parameter apa saja yang ada pada setiap panggilannya.
2. Timer Mocks ialah pengaturan waktu saat melakukan unit testing, dengan mock ini penguji dapat memanipulasi waktu dan mengatur fungsi timer yang berhubungan, seperti: `setTimeout()`, `setInterval()`, `clearTimeout()`, `clearInterval()`.
3. Jest juga menyediakan ES6 Class Mock dengan berbagai cara untuk mock ES6.

### **3.3.1.2 Mengerjakan Unit Testing pada Landing Page**

Tugas selanjutnya yang diberikan adalah membuat Unit Testing pada salah satu file di Landing Page. Dalam melakukan unit testing terdapat coverage yang harus dipenuhi agar sistem berjalan sesuai struktur yang ada, yaitu *Statement*,

*Branches, Functions, Lines. Coverage* yang ada ini harus memenuhi syarat yaitu sekitar 90% - 100% pada setiap komponen yang ada pada *code coverage*. Saat melakukan unit testing biasanya setiap folder yang berisi code mempunyai folder bernama `_test_`. Folder ini adalah bawaan dari tools yang ada di ES6, yaitu babel dan webpack.



Gambar 3.1 Folder test

Dalam folder tersebut terdapat file dengan format `spec.jsx`. File tersebut yang akan digunakan dalam testing salah satu code yang ada. Contohnya saat memanggil header yang akan ditampilkan pada tampilan untuk mobile. Langkah yang akan dibuat adalah mengimport file yang akan diperlukan, file react yang dimana kita memakai framework `reactJS`. Lalu, file yang akan digunakan selanjutnya ialah file source code yang akan di *test*. Setelah mengimport harus melihat properti apa saja yang akan dibutuhkan saat melakukan *testing*, ini dapat



di lihat pada source code filenya. Misalnya, untuk mengetest bagian headernya dibutuhkan property seperti :

- *location* : Berfungsi memanggil link sebagai *root*.
- *getAccountAction* : Berfungsi untuk mengecek apakah sudah masuk dengan akun dan ini akan dilakukan secara berulang sampai dia mengecek seolah-olah sudah masuk dengan akun. Dalam hal ini menggunakan mock dari library jest, yaitu *jest.fn()*. *Jest.fn()* ini berfungsi untuk memberikan nilai kosong untuk memanggil secara berulang-ulang dan mencatat pengulangannya tersebut dan pada parameter mana yang memanggilnya.
- *Title* : Berfungsi memberi judul yang akan ditampilkan pada header.
- *customTitle* : Berfungsi mengisi produk apa yang akan ditampilkan, ini bisa diisi dengan string kosong.

Setelah melakukan testing dengan membuat *function* *test()* berfungsi menampung semua code testing yang akan dijalankan. Menggunakan Jest *library* untuk melihat seberapa coverage source code tersebut. Dapat dilihat dari white box testingnya, apakah sudah mencapai 90% atau belum.

**All files**  
**81.29%** Statements (388/388) **79.67%** Branches (383/383) **81.49%** Functions (388/467) **80.96%** Lines (323/383)

Press n or j to go to the next uncovered block, b, p or k for the previous block.

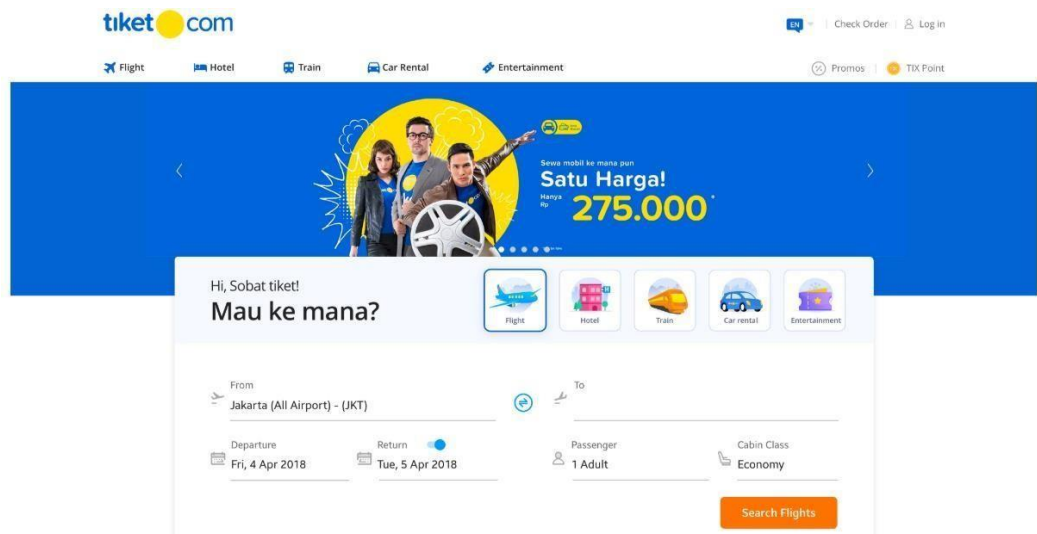
File	Statements	Branches	Functions	Lines
components	100%	18/19	100%	5/5
components/Account	100%	31/31	100%	18/19
components/Alert	100%	4/4	100%	13/13
components/AlertDesktop	100%	3/3	100%	8/8
components/BalanceAmount	100%	16/16	80.32%	28/31
components/BookingCode	100%	10/10	100%	2/2
components/ButtonBar	100%	2/2	100%	0/0
components/BreadCrumbs	100%	4/4	100%	5/5
components/ConfirmationModal	100%	3/3	91.67%	22/24
components/CountdownTimerPanel	100%	25/25	77.78%	14/18
components/CustomSelect	100%	14/14	100%	19/19
components/Drawer	82.14%	23/28	85%	17/20
components/EstimateRefundCard	100%	4/4	100%	8/8
components/FlashMessage	100%	8/8	100%	6/6
components/Flight	100%	25/25	100%	29/29
components/FlightRefundCard	100%	25/25	95.12%	78/82
components/FlightRefundCardV2	100%	11/11	95.12%	39/41
components/Footer	100%	21/21	100%	2/2
components/Header	100%	33/33	97.56%	40/41
components/Header/components/currency	100%	3/3	100%	0/0
components/Hotel	100%	32/32	100%	70/70
components/InputGroup	100%	5/5	100%	0/0
components/InputGroup/components/InputGroupAddon	100%	5/5	100%	0/0
components/InputGroup/components/InputGroupButton	100%	3/3	100%	0/0
components/LabelStatus	100%	5/5	100%	0/0
components/Link	100%	6/6	100%	3/3
components/Loader	100%	5/5	100%	8/8
components/Loading	100%	3/3	100%	0/0
components/ModalV2	100%	8/8	88.88%	21/26
components/NotificationBox	100%	3/3	100%	2/2
components/Passenger	90.36%	151/168	84.47%	174/206
components/Passenger/light	96%	24/25	94.59%	70/74
components/PassengerRefundCard	92.86%	13/14	100%	12/12
components/PassengerV2	95.7%	86/91	82.71%	110/133
components/PeriodRefund	100%	5/5	88.89%	16/18
components/Progressbar	100%	5/5	100%	6/6
components/RefundMethod	100%	11/11	88.36%	38/44
components/RefundProductCard	100%	22/22	78.17%	38/48
components/SelectReason	100%	13/13	85.71%	8/7
components/TermsConditions	80%	20/25	80.00%	25/31
components/TermsConditionsV2	100%	22/22	82.14%	23/28
components/Validator	100%	30/30	100%	4/4
components/WarningCard	100%	5/5	100%	0/0
components/transitions	100%	8/8	100%	0/0
routes/BankAccount	97.5%	39/40	92.31%	36/39
routes/BankAccount/components/BankForm	98.46%	64/65	90.57%	48/53
routes/MyRefund	89.13%	41/46	88.42%	78/114
routes/MyRefund/components/FlightRefund	100%	8/8	83.87%	26/31
routes/MyRefund/components/HotelRefund	100%	4/4	90%	9/10
routes/MyRefund/components/RefundPlaceholder	100%	5/5	100%	3/3
routes/NotFound	100%	1/1	100%	0/0

Gambar 3.2 White box Landing page

### 3.3.1.3 Mengerjakan Header dan Footer pada Landing Page

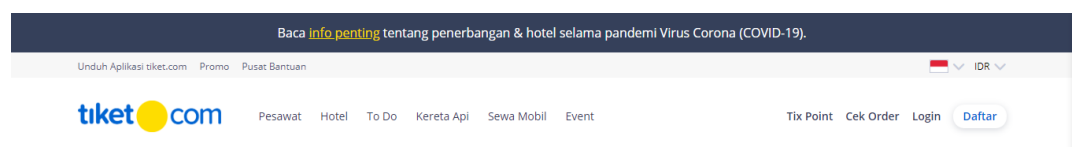
Tugas selanjutnya, tugas yang diberikan ialah mengubah versi *Header* dan *Footer* pada halaman utama. Komponen yang akan berubah seperti ukuran pada

logo dari tiket.com, mengganti *icon* pada menu *product list*-nya, mengubah tata letak header bagian kanan yang terdapat login, tix point dan terdapat penambahan banner notifikasi.



Gambar 3.3 Tampilan Header versi Sebelumnya pada Halaman Utama

Dalam mengubahnya untuk melalui graphql yang menampung semua data dari *backend*. Dalam versi terbaru terlihat perbedaan yang sangat signifikan, seperti logo tiket.com dan produk list berada sejajar.



Gambar 3.4 Tampilan Header versi Terbaru pada Halaman Utama

Jika di klik pada icon bendera Indonesia, akan menampilkan bahasa apa yang ingin digunakan. Tiket.com menyediakan 2 bahasa, yaitu : bahasa Indonesia dan bahasa Inggris.



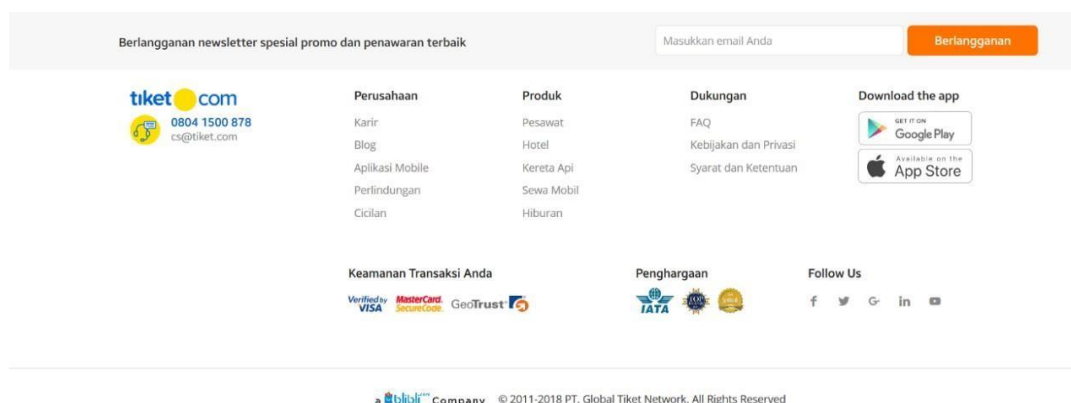
Gambar 3.5 Tampilan *icon* bendera

Sedangkan pada icon „IDR' berfungsi mata uang apa yang akan digunakan oleh *user* untuk sekarang hanya tersedia dalam mata uang Indonesia saja.

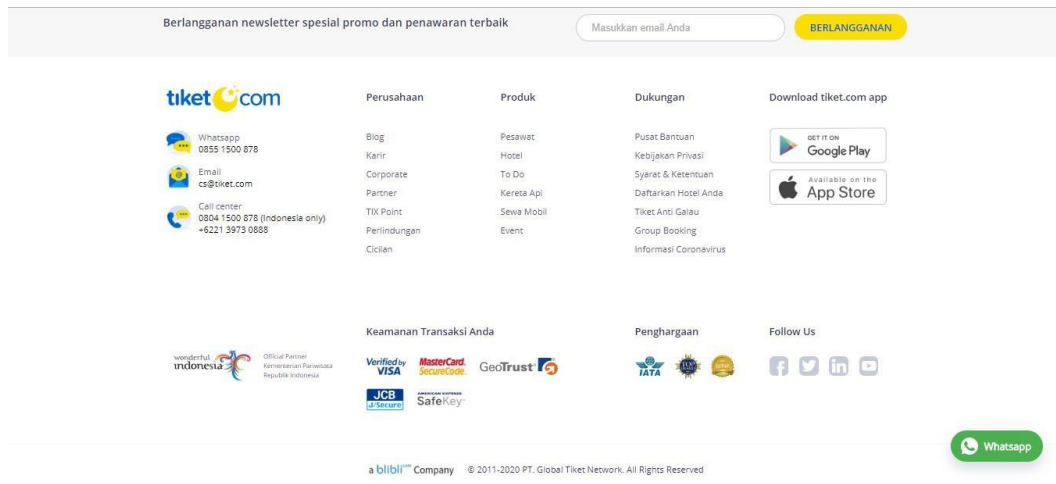


Gambar 3.6 Tampilan *icon* IDR

Pada *footer* yang ada di halaman utama, terlihat perubahan terhadap bentuk dan warna *button* berlangganan menjadi lebih cerah. Selain itu juga terdapat beberapa informasi tambahan seperti, penambahan kontak informasi tiket.com, penambahan transaksi dan lain sebagainya.



Gambar 3.7 Footer Sebelumnya Pada Halaman Utama



Gambar 3.8 Footer Sesudah Pada Halaman Utama

#### 3.3.1.4 Mengerjakan Header dan Footer pada Info Page

Pekerjaan terakhir yang diberikan, ialah mengubah versi header dan footer pada *Info page*. *Header* yang terdapat pada *info page* sedikit berbeda dengan *Landing page*. Pada halaman ini menampilkan informasi promo yang dipilih.



Gambar 3.9 Header Pada Info Page

#### 3.3.1 Kendala yang ditemukan

Dalam proses kerja magang yang dijalani selama 3 bulan, terdapat beberapa kendala-kendala yang ditemukan antara lain:

##### a) Kendala Teknis

Kendala yang dihadapi pertama kali adalah belum adanya pengetahuan akan *framework* dan *tools* yang akan digunakan selama proses kerja magang terutama untuk ES6 dan juga jest sehingga terjadinya hambatan yang cukup

besar pada proses magang. Perbedaan versi dan masalah konfigurasi juga kerap terjadi yang menyebabkan terganggunya proses kerja magang. Ada beberapa *code* yang tidak bisa dijalankan saat menjalankan unit testing, seperti fungsi DOM. DOM ini ialah mengambil waktu dan tanggal saat itu jadi sulit untuk di implementasikan pada unit testing jest, sehingga menyebabkan *coverage* pada unit testing tidak mencapai 90%.

b) Kendala Non Teknis

Kendala lainnya yang dihadapi adalah kendala non teknis seperti malu bertanya pada awal proses kerja magang. Perangkat yang digunakan sering terjadinya *error* misalnya seperti layar yang tiba-tiba berjalan sendiri tanpa ada yang menjalankannya.

### 3.3.2 Solusi atas Kendala yang Ditemukan

Untuk menghadapi masalah teknis, dilakukan pembelajaran lebih intens mengenai *framework* dan *tools* yang digunakan, baik itu melalui pertanyaan-pertanyaan kepada pembimbing ataupun kepada *frontend* tim lain jika pembimbing magang sedang terlihat sibuk yang lebih mengetahui maupun dengan mempelajari menggunakan media internet. Solusi untuk meningkatkan *coverage* ialah menggunakan *comment istanbul ignore next* pada .js, ini berfungsi melewati step tersebut saat melakukan unit testing.

Sedangkan untuk masalah non teknis, dilakukan proses tanya jawab selama 1 jam untuk membahas masalah *frontend* yang dimana semua anak magang dan pekerja tetap membahas apa yang tidak dimengerti atau kendala yang dihadapi. Solusi dari masalah perangkat yang error, telah dibantu oleh *IT Support* dalam mengatasinya dan menemukan penyebab terjadinya error pada perangkat yang digunakan.

