

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Penulis menjalankan program magang dengan posisi *Quality Assurance* di departemen *Product Operations* di PT. Global Loyalty Indonesia. Divisi QA memiliki bentuk koordinasi sehari-hari dengan menggunakan aplikasi Telegram sebagai metode komunikasi baik sesama QA atau pun komunikasi dengan divisi-divisi lain yang berada di dalam project yang sama.

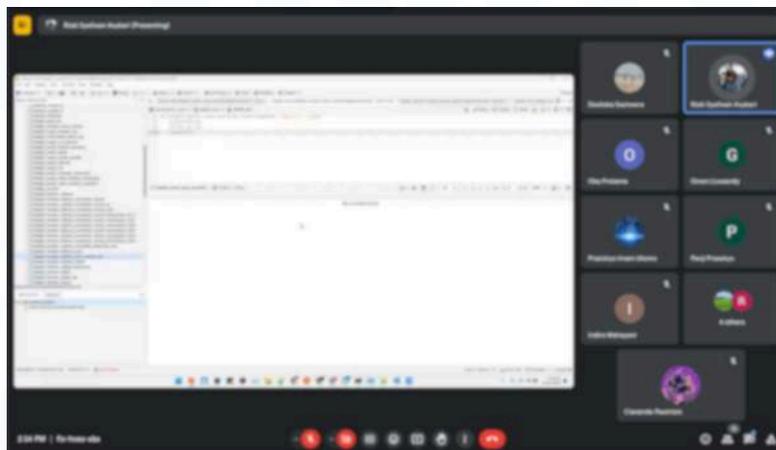
3.1.1. *Daily Update*

Pertemuan yang dilakukan setiap pagi untuk melakukan *update* tentang *progress* pekerjaan di hari sebelumnya, kendala apa saja yang ditemukan dan jika ada masalah yang bisa didiskusikan dalam project tersebut.

3.1.2. *Refinement*

Pertemuan yang membahas tentang flow project secara keseluruhan dan diikuti oleh berbagai divisi baik dari sisi bisnis dan *developer*. Dalam pertemuan ini, biasanya terjadi diskusi antara divisi bisnis dan divisi IT tentang menentukan seberapa sulit project ini dijalankan dan apakah membutuhkan effort yang lebih dari divisi developer.

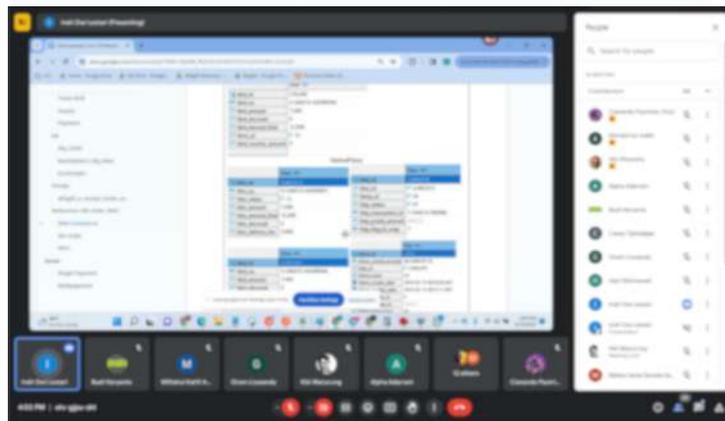
3.1.3. *Sprint Planning*



Gambar 3.1 Sprint

Pertemuan yang dilakukan saat *project* baru dimulai, biasa di *Sprint* 1 QA akan melakukan pertemuan dengan *developer* dan *product development team* untuk membahas *business requirement*, *system design* dan *storylist* sebuah *project* dan membahas tentang rencana *project* yang akan dijalankan. Dalam durasi sebuah *project* berjalan, biasanya terdapat 2 - 4 *Sprint* yang akan dilakukan tergantung dari skala *project* yang berjalan. *Project* yang lebih besar dan kompleks biasanya akan memiliki lebih dari 3 *Sprint* untuk melakukan *update progress* secara berkala dengan semua divisi yang terlibat.

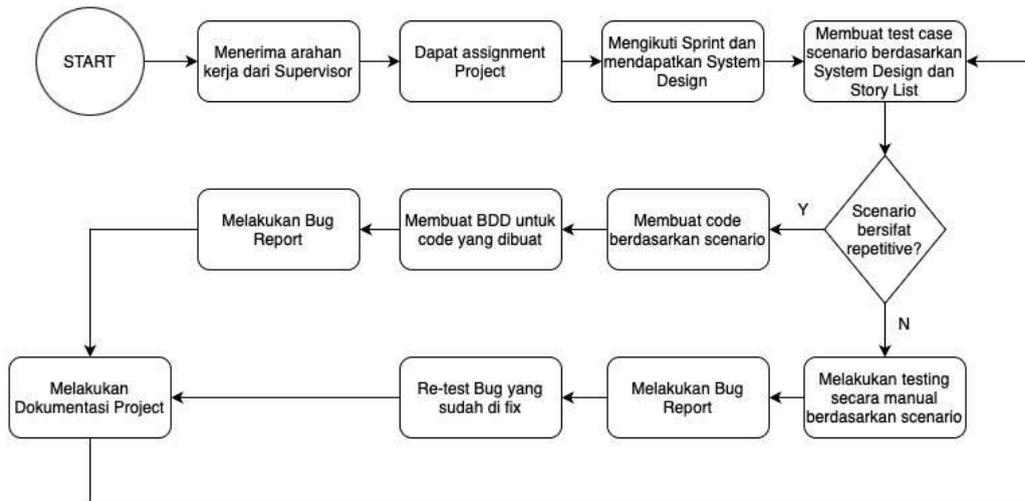
3.1.4. *Sharing Session*



Gambar 3.2 *Sharing Session*

Pertemuan yang berupa presentasi tentang *project* yang sudah selesai dilakukan oleh *Person in Charge* (PIC) *project* tersebut, untuk menjelaskan apa saja yang dilakukan pada *project* tersebut dan guna untuk melihat kembali dan berbagi informasi kepada QA yang lain yang tidak terlibat dalam *project* tersebut. Para PIC juga biasanya akan menjawab dan menjelaskan jika ada QA yang lain yang memiliki pertanyaan terkait *project* yang dipresentasikan.

3.2. Tugas dan Uraian Kerja Magang



Gambar 3.3 Flow Kerja QA

3.2.1 Tugas yang Dilakukan

Selama penulis menjalankan magang di PT. Global Loyalty Indonesia, penulis menjalankan posisi *Quality Assurance* di departemen *Product Operations*. Selama menjalankan magang, penulis ditugaskan oleh *supervisor* untuk terlibat secara langsung ke dalam beberapa *project* baik *project* kecil maupun besar. Peran yang dijalankan oleh penulis sebagai seorang *Quality Assurance* adalah untuk melakukan proses *testing* terhadap aplikasi Alfagift dengan tujuan untuk memastikan bahwa aplikasi Alfagift dapat berjalan dengan sempurna dan bebas *bug* atau kecacatan fungsional lainnya sebelum aplikasi dirilis di publik melalui *platform App Store* maupun *Play Store*.

Berdasarkan gambar 3.3, awalan tugas dari penulis adalah mendapat *assignment* dari *supervisor* untuk masuk ke *project* tertentu. Saat sudah masuk ke dalam sebuah *project*, penulis harus mengikuti *refinement* dan *sprint planning*

project yang akan membahas tentang *flow project* dan *scope* dari *project* yang akan dikerjakan. Selanjutnya, penulis akan mendapatkan *system design* dan rancangan tampilan *front-end* dari divisi *UI/UX developer* yang akan menjadi acuan untuk pembuatan *test case scenario*. Setelah aplikasi sudah berhasil naik ke fase *staging*, maka penulis dan QA lainnya dapat memulai proses *testing* baik secara *manual* maupun *automation*. Di dalam sebuah *project*, penulis akan melakukan proses *testing* dan menggunakan *tools-tools* seperti DBeaver, MongoDB, Solr Admin, dan Swagger UI untuk mengakses *database* dan melakukan pengecekan ataupun intervensi, berinteraksi seperti mengirim dan menerima data dari API yang digunakan untuk fitur tertentu, dan melakukan modifikasi ke stok barang di aplikasi atau menambahkan promo ke produk tertentu.

Dalam proses *testing*, semua temuan *bug* atau kecacatan lainnya di dalam aplikasi akan di dokumentasi dan kemudian penulis akan melakukan koordinasi dengan divisi yang terkait baik itu dari divisi *developer backend*, *frontend*, ataupun divisi yang bertanggung jawab atas *service* yang digunakan untuk aplikasi Alfagift. Setelah *bug* berhasil di *fixing* oleh divisi terkait, maka penulis akan melakukan proses *retest bug* untuk memastikan bahwa *bug* tersebut sudah berhasil di *fixing* dan sudah berjalan dengan semestinya. Saat semua *scenario* di dalam suatu *project* sudah berhasil di *test* dan semua temuan *bug* sudah berhasil di *fixing* dan di *retest* oleh QA, maka QA akan melakukan proses dokumentasi agar *project* ini dapat menjadi referensi untuk QA atau *project* yang terkait di masa depan. *Project* tersebut juga akan mendapatkan status QA *passed*, dan kemudian akan diteruskan ke divisi *Testing Operations*.

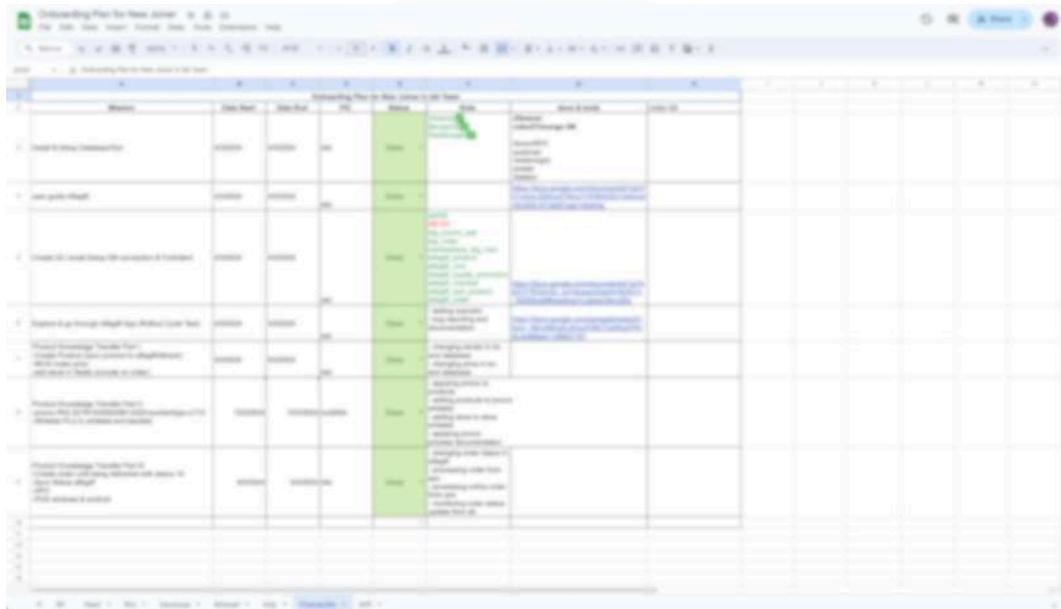
3.2.2 Uraian Kerja Magang

Penulis menjalankan kerja magang mulai dari tanggal 4 Maret 2024 sampai tanggal 20 Mei 2024 di PT. Global Loyalty Indonesia. Pekerjaan yang dilakukan penulis selama menjalankan magang di PT. Global Loyalty Indonesia dapat dilihat di tabel 3.1.

Tabel 3.1 Timeline Kerja Magang

Tugas	Maret				April				Mei			
	1	2	3	4	1	2	3	4	1	2	3	4
Onboarding Plan for New Joiner	■	■										
Full Cycle Rollout		■										
Setup automation testing		■										
Project pengujian fitur Promo Voucher		■	■	■								
Project Flag Promo Toko					■							
Project Remapping Address						■	■	■				
Project Refund Transaksi									■	■	■	■

3.2.2.1 Onboarding Plan for New Joiner



Task	Start Date	End Date	Status	Notes
1. Read & Understand Company Policy	2023-01-01	2023-01-01	Done	
2. Learn about the company	2023-01-01	2023-01-01	Done	
3. Understand the role of the position	2023-01-01	2023-01-01	Done	
4. Understand the company's vision, mission, and values	2023-01-01	2023-01-01	Done	
5. Understand the company's organizational structure	2023-01-01	2023-01-01	Done	
6. Understand the company's products and services	2023-01-01	2023-01-01	Done	
7. Understand the company's competitors	2023-01-01	2023-01-01	Done	
8. Understand the company's market	2023-01-01	2023-01-01	Done	
9. Understand the company's customers	2023-01-01	2023-01-01	Done	
10. Understand the company's employees	2023-01-01	2023-01-01	Done	

Gambar 3.4 Onboarding Plan for New Joiner

Program magang di PT. Global Loyalty Indonesia berdurasi selama 6 bulan dan selama 6 bulan, penulis mengikuti beberapa project yang dikerjakan. Pada hari pertama magang di PT. Global Loyalty Indonesia, penulis mendapatkan mentor sebagai pembimbing sekaligus yang akan mengajarkan cara kerja di posisi QA. Hal-hal yang harus dipelajari telah di rangkap di dalam *google sheets Onboarding Plan for New Joiner* seperti di Gambar 3. *Sheets* tersebut memiliki *checklist* hal-hal yang harus dipelajari serta waktu mulai dan waktu selesai hal dipelajari. *Sheets* ini membantu agar *supervisor* dapat dengan mudah untuk *tracking progress* anak magang, dan agar dapat melihat anak magang mana yang sudah siap untuk mengikuti project berjalan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.2.2 Full Cycle Rollout

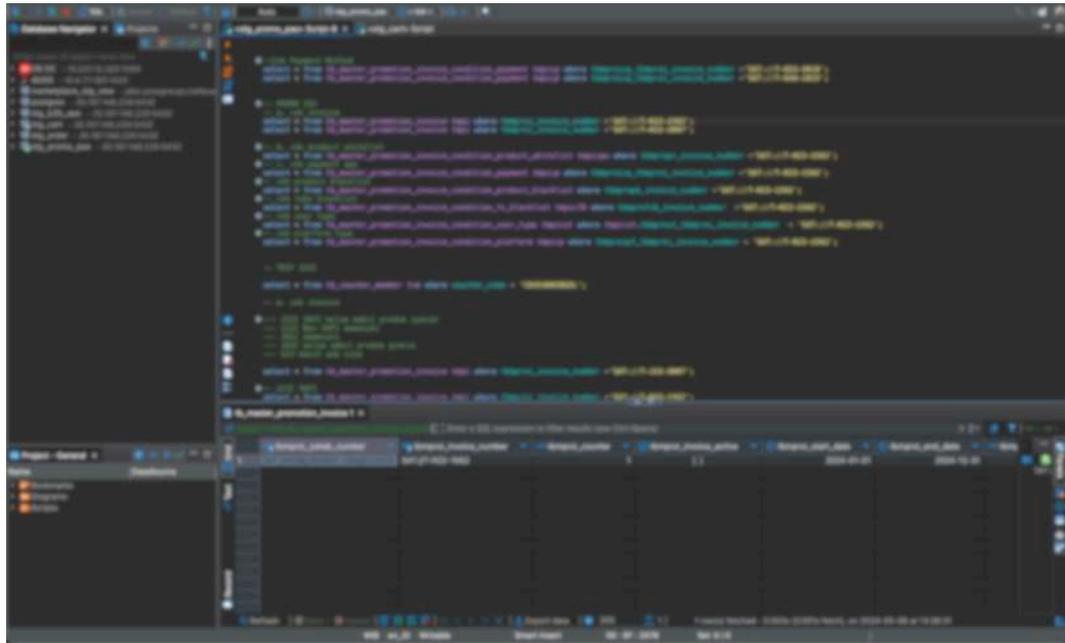
The image shows a screenshot of a software testing tool interface. It features a grid of test cases. The columns include test case ID, description, priority, and status. The status column contains various colored indicators: green, yellow, and orange. The interface also shows a search bar at the top and a list of test cases on the left side.

Gambar 3.5 Full Cycle Rollout

Full Cycle Rollout adalah suatu tahapan untuk pengujian keseluruhan fitur aplikasi Alfragift sebelum versi terbaru aplikasi Alfragift dirilis di Play Store maupun App Store. *Full Cycle Rollout* sebagian besar terdiri dari *positive test case scenario* untuk memastikan semua fitur di dalam aplikasi berjalan dengan baik dan sesuai dengan rancangan, dan untuk memastikan bahwa tidak ada konflik antara satu fitur dengan fitur yang lain.

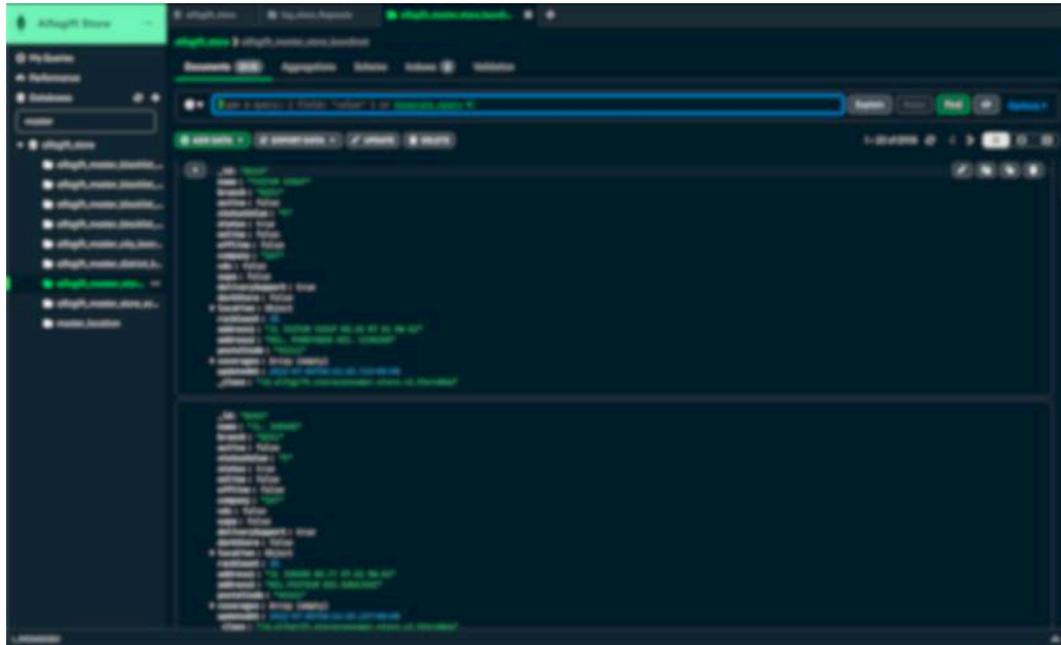
3.2.2.3 Setup tools

Tools atau aplikasi yang digunakan oleh penulis yang juga sering digunakan di divisi QA adalah DBeaver, MongoDB, Solr Admin, Swagger UI. Masing-masing *tools* memiliki fungsi masing-masing memiliki fungsi khusus dan semua *tools* digunakan untuk mengakses database, mengirim dan menerima data dari API, dan lain-lain.



Gambar 3.6 DBeaver

DBeaver adalah aplikasi yang digunakan untuk mengakses macam-macam *database* seperti MySQL, PostgreSQL, MariaDB, SQLite, Apache Family, etc [8]. DBeaver digunakan penulis untuk mengakses *database* dengan tipe PostgreSQL dan memiliki tampilan seperti di Gambar 3.6. DBeaver digunakan karena mudah dipahami dan dapat menyambungkan ke beberapa *database* sekaligus dan dapat melakukan operasi CRUD (*Create, Read, Update, Delete*) dengan mudah dan cepat. DBeaver juga memiliki fitur untuk menyimpan *query* yang sering digunakan agar dapat digunakan dengan berulang tanpa harus menghafalkan *query* tersebut.



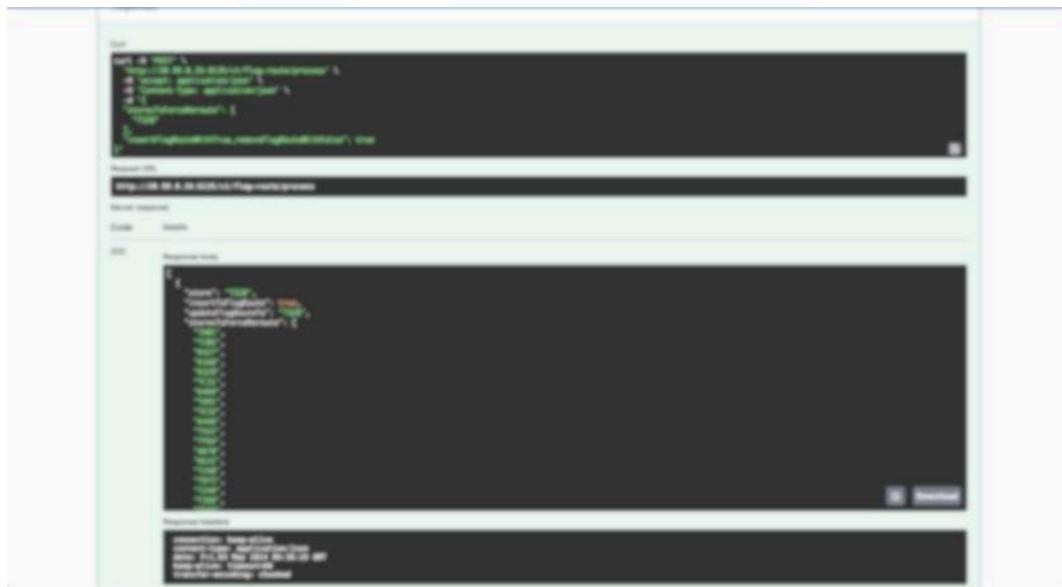
Gambar 3.7 MongoDB Compass

MongoDB Compass adalah aplikasi yang memiliki *Graphical User Interface* (GUI) untuk mengakses *database* berjenis MongoDB yang mudah digunakan dan memiliki tampilan yang interaktif dan *user-friendly* [9]. MongoDB Compass juga memiliki banyak fitur yang mempermudah mengakses data seperti *built-in query bar* yang memudahkan pengguna untuk mencari data spesifik tanpa perlu menulis query yang rumit. MongoDB Compass juga memiliki opsi untuk menunjukkan *schema* dari sebuah *database* yang mempermudah untuk memvisualisasi hubungan antar data yang berada di dalam suatu *database*.



Gambar 3.8 Solr Admin

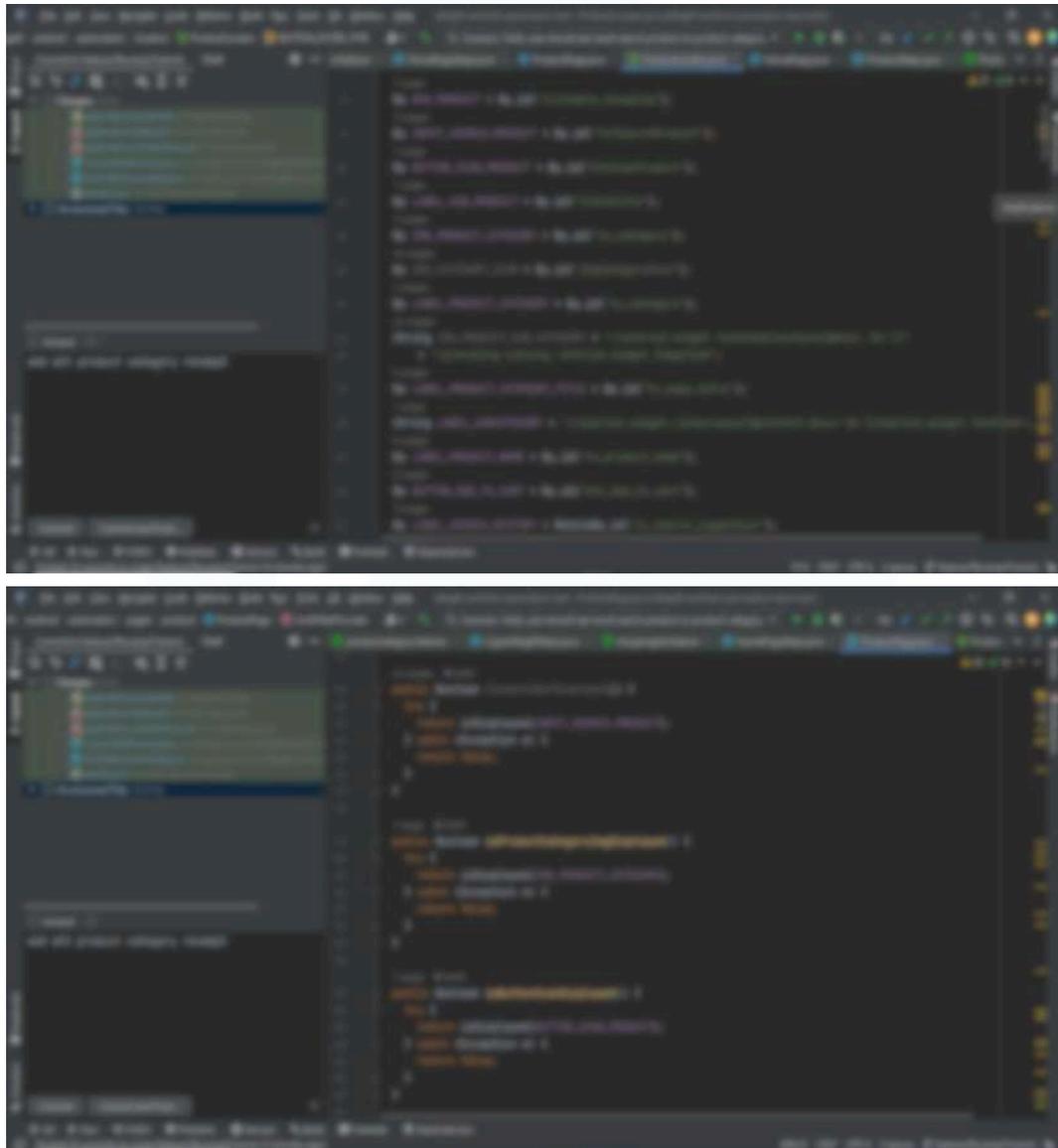
Solr Admin adalah web interface dari Apache Solr yang dapat digunakan untuk memodifikasi, mengelola, dan mengkonfigurasi instance Solr [10]. Di dalam project yang dikerjakan penulis, Solr Admin digunakan untuk melakukan *update* harga atau dapat mengaktifkan barang yang sedang *disabled* atau tidak bisa di *add-to-cart*.



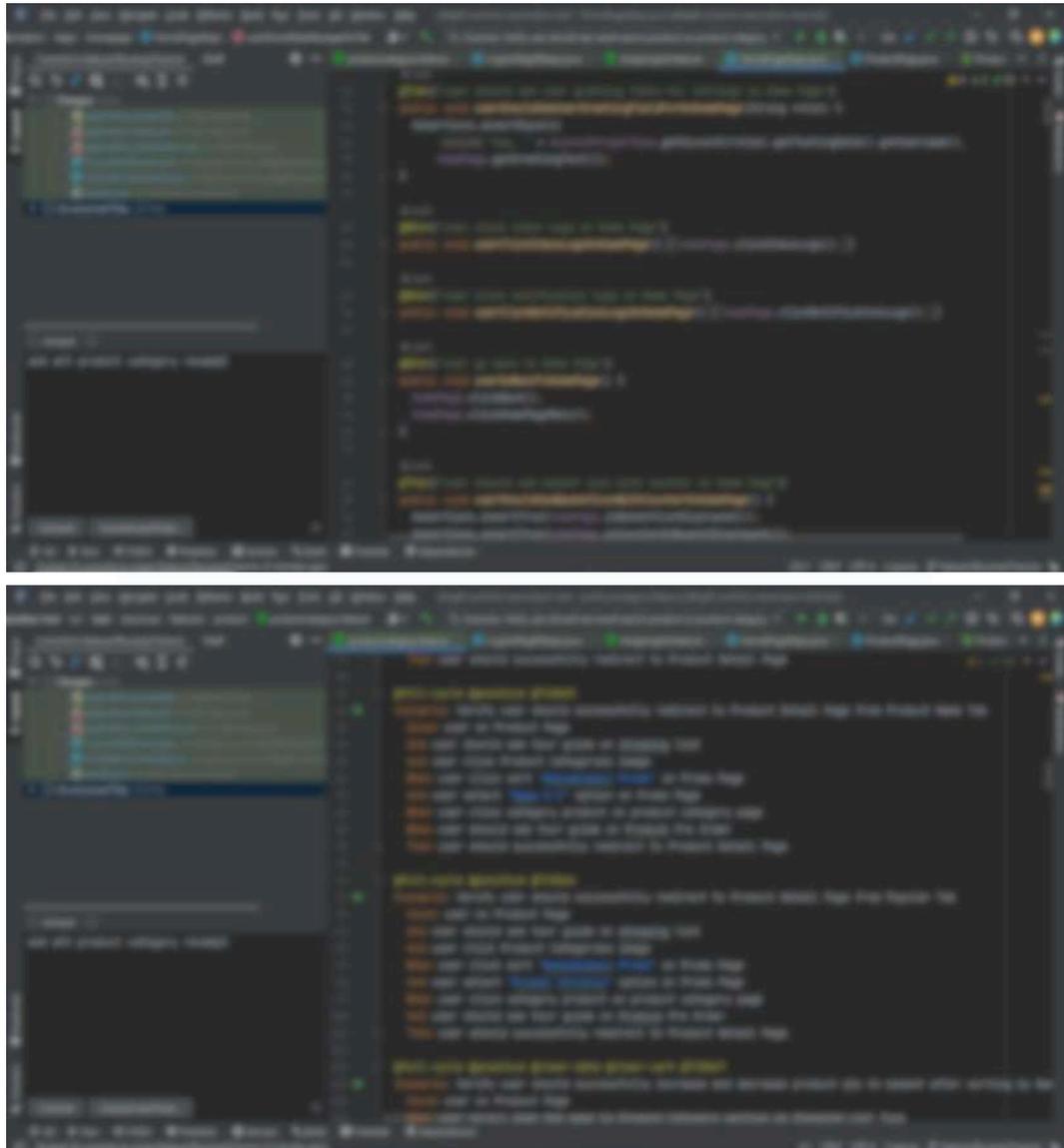
Gambar 3.9 Swagger UI

Swagger UI digunakan oleh penulis untuk proses pengiriman dan penerimaan data kepada API yang digunakan [11]. Swagger UI juga membantu agar penulis dapat memeriksa *JSON response* dari API untuk mengetahui apakah *update* telah dilakukan dan jika proses tersebut berhasil atau gagal.

3.2.2.4 Setup automation testing



UNIVERSITAS
MULTIMEDIA
NUSANTARA



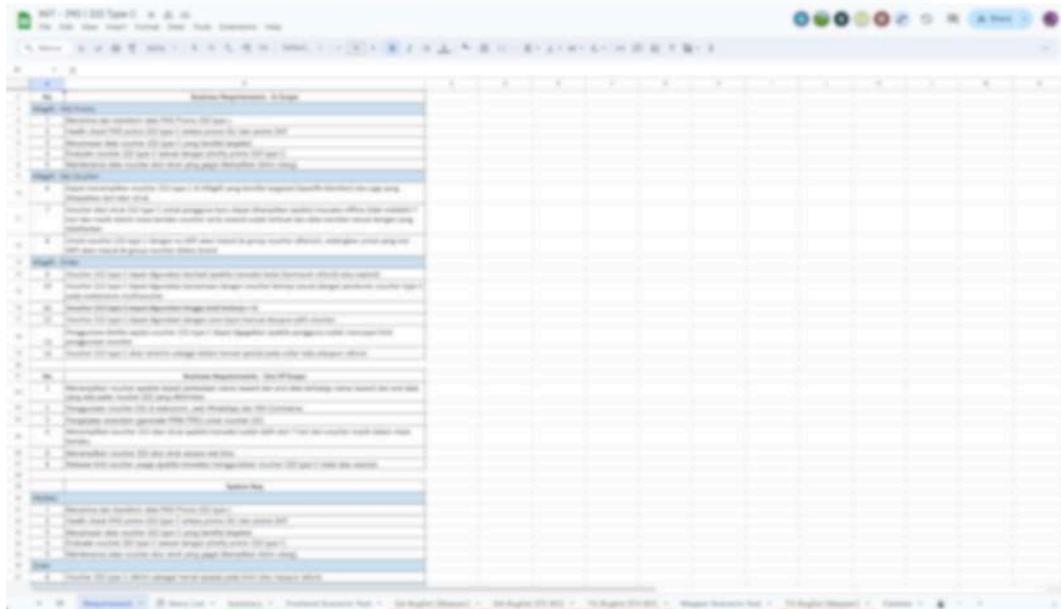
Gambar 3.10 Setup automation testing Appium

Setup automation testing dilakukan pada aplikasi Appium dengan menggunakan bahasa Java untuk melakukan pengujian test case scenario yang bersifat repetitive. Tampilan aplikasi Appium dapat dilihat di gambar 3.10. Tipe *scenario* yang digunakan dalam proses testing *automation* adalah scenario yang bersifat *repetitive*, dimana *scenario* tersebut merupakan *scenario* pengulangan dari *project* yang telah selesai sebelumnya. Setiap proses *automation* selesai, akan muncul *report* dari aplikasi Appium yang akan menunjukkan *success rate* yang

menandakan berapa *scenario* yang berhasil dan yang gagal. *Scenario* yang gagal akan memiliki penyebabnya dan akan dimasukkan ke *buglist* untuk proses *fixing*.

3.2.2.5 Project pengujian fitur Promo *Voucher*

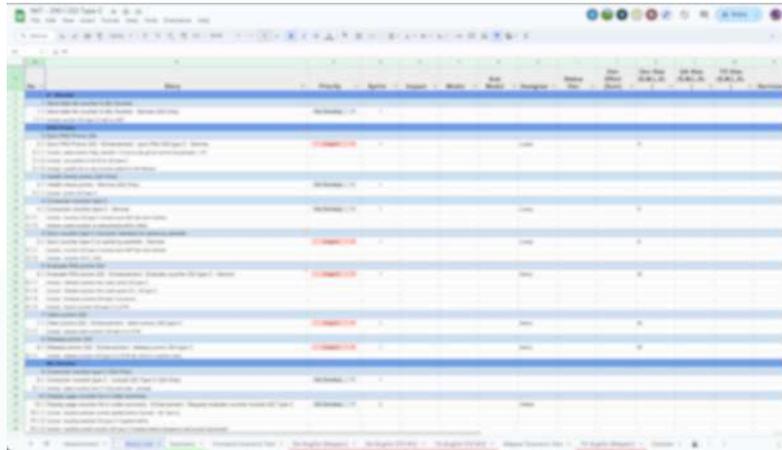
Project Promo *Voucher* adalah *project* yang memunculkan fitur baru dimana promo berbentuk *voucher* yang dapat di *apply* oleh pengguna aplikasi Alfagift. *Project* ini juga memastikan bahwa Promo *Voucher* ini dapat digunakan bersamaan dengan promo-promo yang lain, selama pengguna telah memenuhi syarat masing-masing promo.



Gambar 3.11 *Business Requirement*

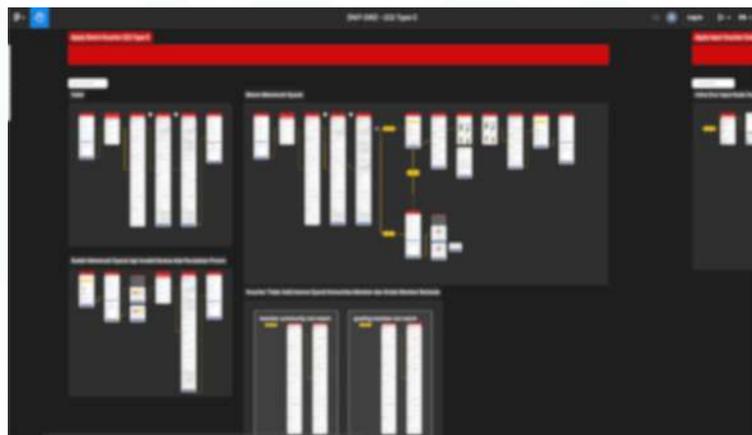
Business requirement membantu memperjelas tujuan, kebutuhan dan ekspektasi di dalam suatu *project*, dengan mendefinisikan tujuan utama dari *project* tersebut. *Business requirement* juga menjadi referensi untuk pengembangan produk di dalam suatu *project* baik fitur maupun konten yang baru. Hal ini juga membantu mengatur *scope* dari *project* tersebut, agar *project* dapat diselesaikan dengan waktu yang telah ditentukan. *Business requirement* juga

dapat menjadi referensi dan informasi tambahan saat membuat *test case scenario* dalam *project*.



Gambar 3.12 *Story List*

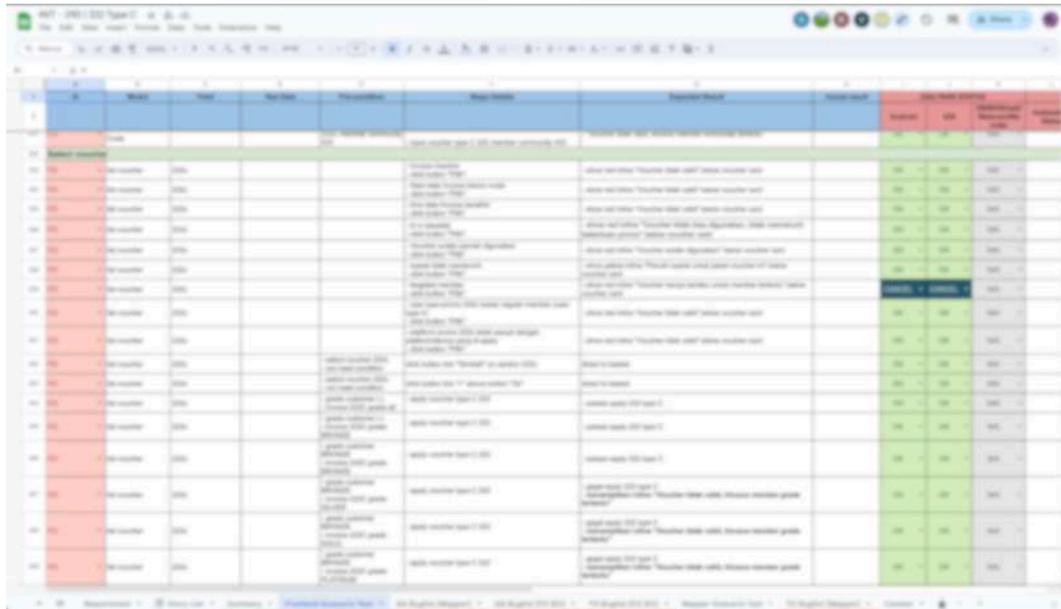
Story list membantu memperjelas fungsionalitas sebuah fitur dari perspektif pengguna, dengan memecahkan sebuah hal yang kompleks menjadi bagian-bagian yang lebih mudah dikelola. *Story list* juga menjadi salah satu acuan dalam membuat *test case scenario* di dalam suatu *project*. *Story list* juga dapat menjadi validasi apakah tujuan bisnis dan ekspektasi pengguna sudah selaras.



Gambar 3.13 Tampilan *front-end*

Tampilan front-end adalah hasil rancangan dari *UI/UX developer* yang akan menjadi acuan saat pembuatan *test case scenario* khususnya yang untuk

bagian *front-end* aplikasi. Hal ini juga membantu memperjelas *flow* dari aplikasi dan dapat menentukan *expected output* saat membuat *positive test case* maupun *negative test case*.



ID	Title	Description	Steps	Expected Output	Status
TC-001	Validasi Email	Memeriksa apakah email yang dimasukkan valid.	1. Masukkan email yang valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Email valid'.	OK
TC-002	Validasi Password	Memeriksa apakah password yang dimasukkan memenuhi syarat.	1. Masukkan password yang tidak memenuhi syarat. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Password tidak valid'.	OK
TC-003	Validasi Nomor HP	Memeriksa apakah nomor HP yang dimasukkan valid.	1. Masukkan nomor HP yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Nomor HP tidak valid'.	OK
TC-004	Validasi Nama	Memeriksa apakah nama yang dimasukkan valid.	1. Masukkan nama yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Nama tidak valid'.	OK
TC-005	Validasi Tanggal Lahir	Memeriksa apakah tanggal lahir yang dimasukkan valid.	1. Masukkan tanggal lahir yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Tanggal lahir tidak valid'.	OK
TC-006	Validasi Alamat	Memeriksa apakah alamat yang dimasukkan valid.	1. Masukkan alamat yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Alamat tidak valid'.	OK
TC-007	Validasi Kode Pos	Memeriksa apakah kode pos yang dimasukkan valid.	1. Masukkan kode pos yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Kode pos tidak valid'.	OK
TC-008	Validasi Jenis Kelamin	Memeriksa apakah jenis kelamin yang dimasukkan valid.	1. Masukkan jenis kelamin yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Jenis kelamin tidak valid'.	OK
TC-009	Validasi Pekerjaan	Memeriksa apakah pekerjaan yang dimasukkan valid.	1. Masukkan pekerjaan yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Pekerjaan tidak valid'.	OK
TC-010	Validasi Pendidikan	Memeriksa apakah pendidikan yang dimasukkan valid.	1. Masukkan pendidikan yang tidak valid. 2. Klik tombol 'Daftar'.	Menampilkan pesan 'Pendidikan tidak valid'.	OK

Gambar 3.14 *Test case scenario*

Setiap *project* akan memiliki *google sheets* yang memiliki daftar *test case scenario*. Setiap *test case* yang dibuat akan memiliki *precondition* atau kondisi yang harus dipenuhi sebelum menjalankan *steps* untuk menguji *scenario* tersebut. Setiap *scenario* juga akan memiliki *steps details* yang berisi cara menguji *scenario* tersebut dengan *detail* dan mudah dipahami dan akan mempermudah proses *testing*. Setiap *scenario* juga akan memiliki *expected output*, untuk menjadi pembandingan apakah hasil *testing* sudah sesuai dengan *expected output*, dan jika memang sudah sesuai, maka *scenario* tersebut dapat diberikan status OK. Setiap *scenario* yang dibuat oleh QA akan melalui pengujian di berbagai *platform* seperti aplikasi Android, IOS, dan Web Commerce (WebCom). Project pengujian fitur promo *voucher* lebih banyak menggunakan proses *testing manual* karena *project* bersifat fitur baru.

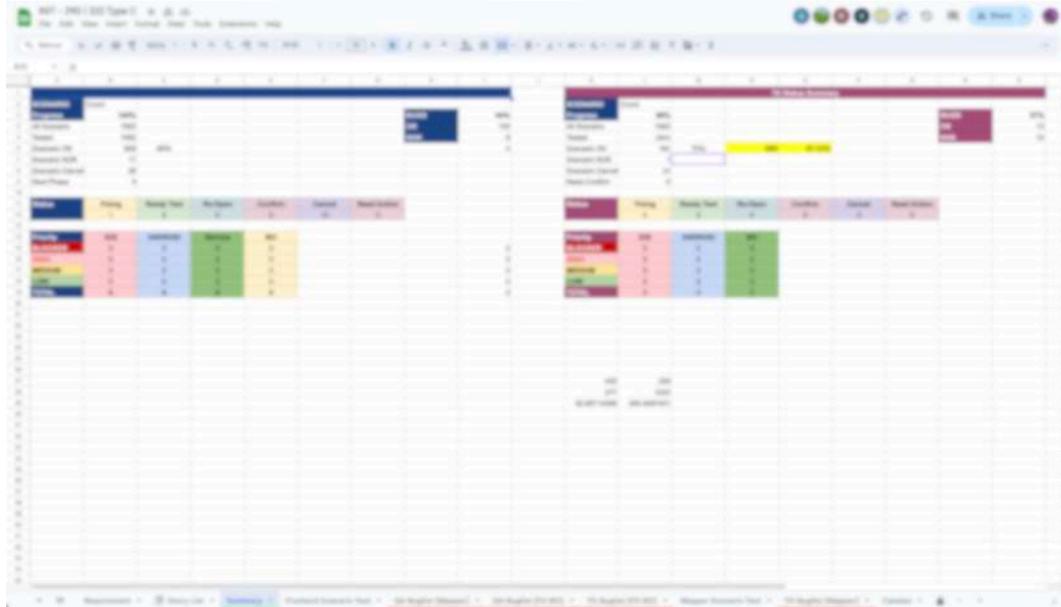
ID	Status	Priority	Description	Assignee																	
1	Open	High
2	Open	Medium
3	Open	Low
4	Open	High
5	Open	Medium
6	Open	Low
7	Open	High
8	Open	Medium
9	Open	Low
10	Open	High

Gambar 3.15 Buglist QA

Selama proses pengujian atau *testing case scenario*, semua *bug* yang ditemukan akan dicatat di *google sheets bug list QA*, dan setiap *bug* yang dicatat akan memiliki *step details* untuk memudahkan proses *recreate bug*, kemudian akan ada *screenshot* dari *bug* yang ditemukan, dan setiap *bug* akan memiliki status QA yang akan menandakan bahwa *bug* sudah di *fix* atau belum, dan ada juga status *Developer* yang akan menandakan bahwa *bug* sudah di *fixing* atau belum. Setiap *bug* juga akan memiliki *priority* seperti *low*, *medium*, *high*, dan *blocker*. Sesuai dengan nama masing-masing *priority*, *low* berarti memiliki *priority* yang cukup rendah dan tidak terlalu berdampak kepada performa aplikasi dan proses *testing*, dan *blocker* adalah *priority* tertinggi yang harus segera di *fixing* karena menghambat proses *testing*.

Salah satu contoh penemuan *bug* adalah ketika menguji *scenario* dimana sebuah *voucher A* memiliki syarat produk A. Saat melakukan proses *checkout* produk A, *voucher A* masih tidak bisa di *apply*, dan *expected case* nya adalah *voucher A* bisa di *apply*. Penemuan *bug* ini kemudian dimasukkan ke dalam *buglist* sebagai salah satu temuan dan diberi *status priority high*, dan setelah *bug*

tersebut telah di *fixing*, maka dilakukan pengujian lagi hingga *scenario* sudah dapat berjalan dengan baik.



Gambar 3.16 *Project Summary*

Setiap *project* yang dikerjakan akan memiliki *google sheets* yang bernama *project summary*. *Project summary* berisi data-data seperti jumlah *test case scenario* yang berada di dalam suatu *project*, jumlah *scenario* yang sudah di *test*, jumlah *scenario* yang sudah memiliki status OK, dan persentase yang menunjukkan jumlah keseluruhan *scenario* yang sudah melewati proses testing. *Project summary* memudahkan untuk *QA tracking* sebuah *project* apakah berjalan dengan lancar dan sesuai *timeline* yang telah ditentukan dan dapat dengan mudah memperkirakan waktu berapa lagi yang dibutuhkan sampai *project* selesai.

3.2.3 Kendala yang Ditemukan

- a. Menggunakan tools-tools baru yang belum pernah digunakan (MongoDB, Solr Admin, Swagger UI, DBeaver)
- b. Kurang pengalaman tentang project yang kompleks
- c. Banyak hal baru yang harus dipelajari dalam waktu singkat

3.2.4 Solusi atas Kendala yang Ditemukan

- a. Membaca dokumentasi resmi *tools-tools* tersebut untuk mencari tahu cara kerja aplikasi tersebut. Membaca *tutorial* yang berada di internet juga sangat membantu penulis lebih memahami cara menggunakan *tools-tools* yang sebelumnya belum pernah digunakan. Bertanya kepada karyawan tetap di divisi *Quality Assurance* tentang cara menggunakan aplikasi tersebut juga membuat penulis lebih paham.
- b. Berdiskusi dengan QA yang lain ataupun *developer* agar lebih paham tentang konsep-konsep di dalam *project*.
- c. Membuat catatan saat belajar tentang cara-cara melakukan intervensi *database* dan melakukan *query*, agar dapat mereview kembali setelah mendapat penjelasan dari mentor maupun rekan kerja. Mencatat *step-by-step* proses yang harus diingat agar tidak perlu bertanya kepada mentor setiap kali ditugaskan untuk melakukan sesuatu yang sudah pernah diajarkan.