

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama periode magang *Track* yang pertama, dalam divisi *Quality Assurance* (QA) di bawah naungan Bapak Benny Chandra yang memimpin departemen *Product Operation*, berperan sebagai anggota tim magang. Departemen *Product Operation* merupakan bagian integral dari struktur organisasi PT Global Loyalty Indonesia, yang fokus pada pengembangan dan pemeliharaan aplikasi *mobile* Alfacift serta aspek teknis lainnya yang mendukung operasional perusahaan. Dalam struktur organisasi ini, tindakan sebagai *Quality Assurance intern*, yang secara langsung dibimbing oleh Kak Alpha Adarrani sebagai mentor. Peran mentor sangat penting dalam mengarahkan dan melatih dalam menjalankan tugas-tugas QA dengan efektif dan efisien.

Sebagai anggota tim QA, tentunya akan terlibat dalam berbagai proyek yang sedang dijalankan oleh mentor. Tugas-tugas tersebut mencakup pengujian perangkat lunak, identifikasi masalah atau cacat dalam aplikasi, serta memberikan masukan konstruktif untuk meningkatkan kualitas produk. Dalam menjalankan tugas-tugas ini, nantinya akan berkoordinasi langsung dengan mentor dan anggota tim lainnya dalam tim proyek.

Koordinasi tim dilakukan secara berkala melalui beberapa saluran komunikasi. Pertama, terdapat pembaruan harian (*daily update*) yang dilakukan secara langsung maupun melalui grup chat di platform Telegram, di mana setiap anggota tim memberikan laporan tentang kemajuan tugas-tugas mereka serta memperoleh arahan baru dari mentor. Selanjutnya, ada juga pertemuan mingguan (*weekly meeting*) di mana anggota tim QA berkumpul untuk membahas proyek-proyek yang sedang berjalan, menyelesaikan masalah yang mungkin muncul, dan merencanakan langkah-langkah selanjutnya.

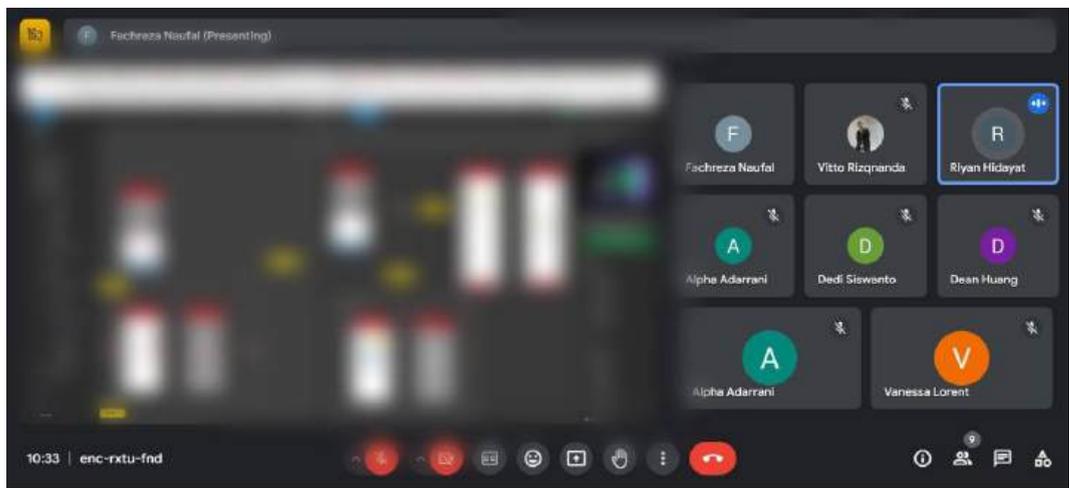
Selain itu, sebagai bagian dari divisi *Quality Assurance*, yang juga terhubung dengan *Supervisor Product Operation* melalui grup WhatsApp khusus. Grup ini bertujuan untuk memfasilitasi komunikasi antara tim QA dan manajemen yang bertanggung jawab atas operasional aplikasi Alfagift serta proyek-proyek terkait. Dengan cara ini, sistem koordinasi yang terstruktur dan beragam memungkinkan tim QA untuk bekerja secara efektif dalam menjalankan tugas-tugas mereka, memastikan kualitas produk yang tinggi, dan menyumbangkan nilai tambah bagi kesuksesan perusahaan secara keseluruhan. Koordinasi kegiatan yang dilakukan oleh tim QA dijabarkan dalam tabel 3.1.

**Tabel 3.1 Koordinasi Kegiatan Tim QA**

<b>Kegiatan</b>	<b>Frekuensi Kegiatan</b>	<b>Tujuan</b>
<i>Daily Update</i>	Setiap Hari	<ul style="list-style-type: none"> <li>- Memberikan laporan tentang aktivitas yang telah dilakukan pada hari sebelumnya.</li> <li>- Memberikan gambaran tentang rencana kerja atau tugas yang akan dilakukan pada hari tersebut.</li> <li>- Membahas hambatan atau masalah yang muncul selama pelaksanaan proyek.</li> </ul>
<i>Weekly Meeting</i> (tim QA)	Setiap hari Rabu	<ul style="list-style-type: none"> <li>- Memberikan pembaruan terhadap kemajuan proyek yang sedang dikerjakan oleh setiap anggota divisi <i>Quality Assurance</i>.</li> <li>- Mengadakan diskusi tentang rencana kerja mingguan yang mencakup tugas-tugas yang akan dilakukan oleh anggota tim QA selama minggu tersebut.</li> </ul>
<i>Sharing Session</i> by tim <i>Human Capital</i>	Setiap hari Sabtu	<ul style="list-style-type: none"> <li>- Melakukan update pada proyek yang sedang dikerjakan, berkonsultasi apakah ada kesulitan atau hambatan selama seminggu.</li> </ul>

Kegiatan	Frekuensi Kegiatan	Tujuan
<i>Sprint</i>	Setiap proyek baru hendak dimulai	<ul style="list-style-type: none"> <li>- Melakukan analisis dan pembahasan mengenai desain sistem yang diterapkan dalam proyek yang sedang dikerjakan.</li> <li>- Mengadakan diskusi tentang cakupan atau ruang lingkup pekerjaan yang harus dilakukan oleh setiap divisi dalam proyek tersebut.</li> </ul>
<i>Glitz Assemble</i>	Pada akhir bulan/ <i>event-event</i> tertentu	<ul style="list-style-type: none"> <li>- Memperoleh informasi terkini mengenai perkembangan dan aktivitas yang terjadi di seluruh perusahaan.</li> <li>- Menggalang upaya untuk meningkatkan semangat dan motivasi para karyawan di berbagai departemen perusahaan.</li> </ul>

### 3.1.1 Daily Update



**Gambar 3.1** Gambaran *Daily Update*

*Daily update* adalah sesi pembaruan kerja yang melibatkan seluruh tim dalam sebuah proyek, termasuk tim *Service*, *Backend*, *Mobile*

*Developer, Project Development*, maupun QA. Sesi ini biasanya diselenggarakan secara langsung di kantor atau dapat melalui *Google Meet*, dimulai dengan laporan tentang pekerjaan yang telah dilakukan oleh setiap anggota tim pada hari sebelumnya serta perencanaan untuk hari tersebut. Selain itu, dalam sesi ini juga dibahas kendala atau masalah yang ditemui oleh tim QA, termasuk *bug* atau cacat yang telah diidentifikasi. Ini adalah kesempatan bagi tim untuk berbagi informasi, mengkoordinasikan upaya mereka, dan mengatasi hambatan yang mungkin muncul dalam proyek. Dengan melakukan *daily update* secara teratur, tim dapat tetap terhubung, memastikan bahwa proyek berjalan sesuai rencana, dan menyelesaikan masalah dengan cepat untuk menjaga kemajuan proyek.

### **3.1.2 Weekly Meeting**

Pertemuan mingguan, sesuai dengan namanya, adalah pertemuan yang diadakan sekali seminggu untuk mendapatkan *update* dari setiap anggota tim. Tujuan utama pertemuan ini adalah untuk membahas topik-topik terkait proyek yang sedang berlangsung dan merencanakan kegiatan mendatang. Pertemuan ini bisa diadakan secara tatap muka di ruang rapat atau secara daring menggunakan *platform* seperti *Google Meet*, tergantung pada ketersediaan ruang dan anggota tim. Fokus utama dari pertemuan mingguan adalah untuk memastikan komunikasi dan koordinasi yang efektif di antara seluruh anggota tim, memastikan bahwa semua orang memiliki pemahaman yang jelas tentang proyek dan peran mereka, serta untuk mengidentifikasi dan menyelesaikan setiap masalah atau tantangan yang muncul selama pengembangan proyek.



**Gambar 3.2 Weekly Meeting Tim QA**

Pada gambar 3.2 diatas menampilkan Tim QA yang sedang melakukan *weekly meeting*. Pada meeting ini tim QA secara bergantian memberikan *update* terkait *project-project* yang sedang dijalankan.

### **3.1.3 Sharing Session by Tim Human Capital**

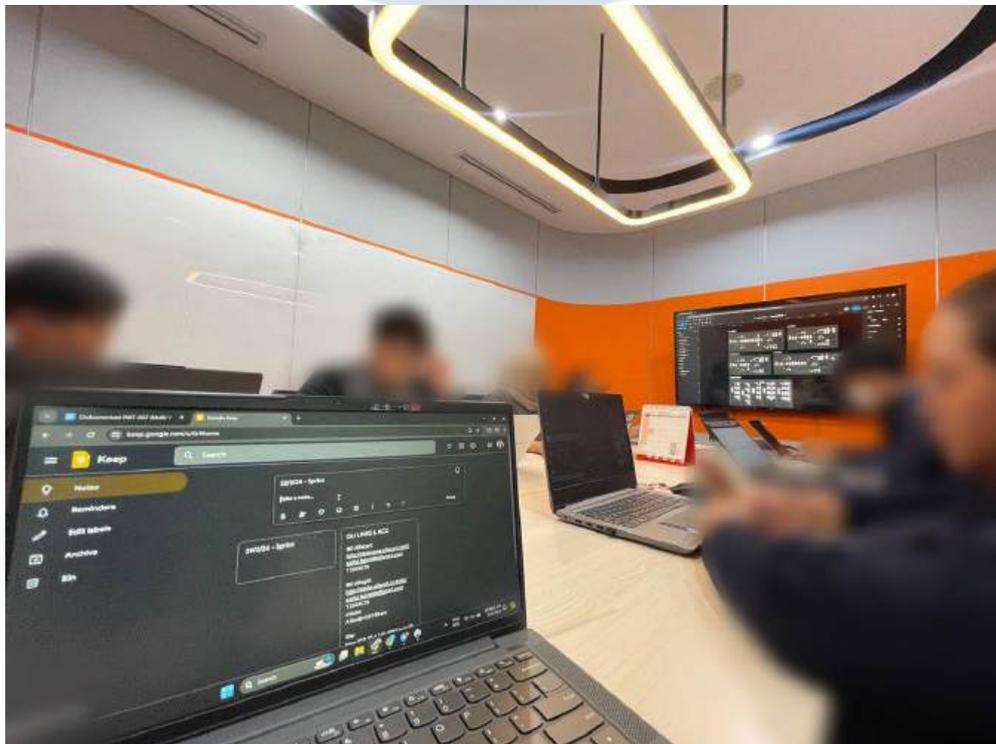
Kegiatan *sharing session* oleh tim *Human Capital* yang dilakukan setiap hari Sabtu bertujuan untuk memberikan masukan atau kabar tentang proyek yang sedang dikerjakan serta memberikan kesempatan kepada setiap anggota tim untuk berkonsultasi mengenai segala kesulitan atau hambatan yang mereka alami selama seminggu tersebut.

Kegiatan ini dijalankan untuk semua anak magang yang berada di PT Global Loyalty Indonesia (GLI) dan diimplementasikan secara *hybrid*, artinya dapat dilakukan baik secara *onsite* maupun *online*, tergantung pada kebutuhan dan situasi. Bagi mereka yang berada di lokasi kantor, mereka dapat menghadiri sesi secara langsung di tempat, sementara bagi mereka yang berada di lokasi yang jauh atau memilih untuk tetap di rumah,

mereka dapat mengikuti sesi secara daring melalui *platform* video konferensi seperti *Zoom* atau *Google Meet*.

Dalam sesi ini, setiap anggota tim, termasuk para magang, memiliki kesempatan untuk berbagi pengalaman, memberikan pembaruan tentang kemajuan proyek yang sedang dikerjakan, dan bertanya tentang masalah atau kendala yang mereka hadapi. Tim Human Capital juga dapat memberikan saran atau bantuan untuk mengatasi masalah yang muncul dan memastikan bahwa semua anggota tim mendapatkan dukungan yang diperlukan untuk sukses dalam magang mereka. Dengan demikian, kegiatan *sharing session* ini tidak hanya menjadi forum untuk berbagi informasi tetapi juga menjadi wadah untuk memperkuat kolaborasi dan dukungan antar anggota tim

#### 3.1.4 *Sprint*



**Gambar 3.3 Kegiatan *Sprint***

*Sprint* adalah pertemuan yang diadakan 1 hingga 2 kali tergantung pada tahap proyek. Dalam *sprint* ini, akan dibahas cakupan pekerjaan yang harus dilakukan oleh setiap divisi terkait dalam proyek tersebut. Diskusi juga mencakup *timeline* kerja dan upaya yang diperlukan dari masing-masing tim untuk memenuhi jadwal tersebut. Gambar 3.2 memberikan gambaran visual tentang *Sprint* yang dilakukan secara *offline*. Ini adalah kesempatan bagi tim untuk secara efisien menetapkan arah dan prioritas kerja mereka, serta memastikan bahwa setiap divisi memiliki pemahaman yang jelas tentang tugas mereka dalam mencapai tujuan proyek secara keseluruhan. Dengan demikian, *sprint* menjadi alat yang sangat penting dalam mengelola proyek dan memastikan kelancaran jalannya.

### **3.1.5 Glitz Assemble**

*Glitz Assemble* merupakan pertemuan langsung yang diadakan di ruang assemble dan diikuti oleh seluruh tim PT Global Loyalty Indonesia (GLI). Pertemuan ini biasanya dipimpin oleh pimpinan tingkat C untuk membahas pembaruan bulannya serta membahas *goals* yang akan dicapai pada bulan berikutnya. Selain itu, pada kegiatan ini biasanya mengadakan juga sesi seminar atau inspirasi yang diberikan melalui cerita-cerita berdasarkan pengalaman sehingga meningkatkan motivasi. *Glitz Assemble* juga menjadi momen di mana seluruh karyawan GLI berkumpul di satu lantai yang sama, memungkinkan interaksi dan kolaborasi antar tim. Selain acara-acara yang telah disebutkan sebelumnya, *Glitz Assemble* juga kadang-kadang menjadi *platform* untuk menyelenggarakan seminar atau sesi diskusi mengenai topik-topik yang relevan dengan kesejahteraan mental dan kesehatan jiwa, seperti cara mengatasi stres atau menjaga kesehatan mental. Ini bertujuan untuk memberikan pengetahuan dan keterampilan kepada karyawan dalam menghadapi tantangan-tantangan kehidupan sehari-hari dan meningkatkan

kesejahteraan mereka secara keseluruhan. Pada akhir kegiatan, terdapat penutup yang meriah, termasuk perayaan ulang tahun bagi karyawan yang berulang tahun dalam bulan tersebut, penyambutan anggota baru ke dalam tim, dan pembagian makanan untuk meningkatkan suasana kebersamaan.

### 3.2 Tugas dan Uraian Kerja Magang

Program magang di PT. Global Loyalty Indonesia berlangsung selama enam bulan, dimulai pada bulan Februari 2024. Dalam program magang ini, setiap mahasiswa magang akan ditugaskan kepada seorang mentor yang akan membimbing mereka sepanjang program. Mahasiswa magang akan terlibat dalam proyek-proyek yang dipimpin oleh mentor mereka. Bagi mahasiswa magang di bidang *Quality Assurance* (QA), tugasnya akan sesuai dengan deskripsi pekerjaan QA, yang meliputi pengujian perangkat lunak baik secara manual maupun dengan menggunakan alat otomatis. Mahasiswa ini akan terlibat dalam berbagai aspek pengujian produk untuk memastikan kualitasnya sesuai dengan standar yang ditetapkan.



**Gambar 3.4** *Flowchart* kerja magang

Saat menjalani program magang, proses kerja dapat disajikan melalui diagram alir yang tersedia dalam Gambar 3.4. Semua langkah dalam tugas dimulai dengan arahan dan pembagian tugas dari *supervisor*, kemudian akan disatukan dalam

proyek yang sama setelah diarahkan. Proses dimulai dengan mengikuti Sprint untuk memperoleh pemahaman menyeluruh tentang proyek. Sprint diadakan sebanyak 2-4 kali tergantung pada tingkat kompleksitas proyek. Setiap *Sprint* mencakup pembagian sistem desain yang mencakup ruang lingkup, alur fitur, dan konsep desain fitur.

Pembuatan skenario berdasarkan SD yang telah dibagikan, yang terdiri dari skenario tampilan *interface* ataupun bagian dalam seperti sisi *database*. Skenario tersebut disusun menggunakan *Google Sheets* dan mencakup *Requirement*, Alur cerita, Rangkuman berapa persen yang telah dikerjakan, *Scenario Test*, dan QA *Buglist*. Skenario ini menjadi panduan untuk pengujian dan pembuatan kode untuk pengujian otomatis. Pengujian otomatis dilakukan jika proyek bersifat repetitif, menggunakan Appium dengan bahasa pemrograman Java. Dengan membuat kode berdasarkan skenario dan test case yang ada. BDD (*Behavior-Driven Development*) digunakan untuk memudahkan *Project Manager* memahami alur pengujian, dengan menggunakan kata-kata yang mudah dimengerti dan merepresentasikan kode di belakangnya.

Setelah kode dan BDD selesai, dilakukan uji coba untuk memastikan tidak ada gangguan teknis yang menghambat pengujian. *Bug* yang ditemukan dilaporkan dan setelah diperbaiki, dilakukan uji coba ulang untuk memastikan fitur berjalan dengan baik. Jika proyek tidak memerlukan pengujian otomatis, pengujian dilakukan secara manual. Setelah pengujian manual selesai, dilakukan pelaporan *bug*. Setelah *bug* siap untuk diuji kembali, maka selanjutnya akan melakukan pengujian ulang untuk memastikan semua *test case* telah terpenuhi. Terakhir, dilakukannya dokumentasi untuk memastikan informasi mengenai proyek dan perubahan fitur dapat diakses oleh anggota QA di masa mendatang.

### **3.2.1 Tugas Kerja Magang**

Ketika mengikuti program magang di perusahaan Alfagift, tugas sebagai sebagai *Quality Assurance* di bagian *Product Operation*. Sebagai

QA, tanggung jawab utamanya adalah melakukan pengujian pada aplikasi Alflagift, terutama pada fitur-fitur baru sebelum diluncurkan ke publik. Pengujian dilakukan secara manual untuk fitur baru yang tidak repetitif, sementara fitur-fitur yang memerlukan pengujian berkala seperti welcome page, login, masukan ke keranjang, dan lain sebagainya, akan diuji menggunakan metode automation *testing*.

Setiap proyek yang ditangani akan dimulai dengan *sprint* yang membahas gambaran umum proyek dan cakupan kerja masing-masing divisi yang terlibat. Langkah selanjutnya adalah dengan melakukan pembuatan skenario pengujian berdasarkan sistem desain dan UI/UX yang telah dirancang dalam Figma. Setelah aplikasi diperbarui oleh tim pengembang, divisi QA akan memulai proses pemeriksaan aplikasi tersebut.

Selama proses proyek, seorang QA juga memiliki akses dalam melakukan *query* untuk memperbarui stok, memeriksa status pesanan, ataupun menambahkan promo produk, yang semuanya dilakukan melalui *database*, *Swagger*, atau Solr. Hal ini menunjukkan bahwa tugas seorang QA tidak hanya terbatas pada pengujian aplikasi, tetapi juga dapat melibatkan interaksi dengan sistem *backend* untuk memastikan fungsionalitas dan keandalan aplikasi secara menyeluruh.

Setiap kali ditemukan *bug* atau kesalahan dalam aplikasi, akan dibuat dokumen laporan yang dapat diakses oleh seluruh anggota tim proyek. Tim *Quality Assurance* (QA) akan menyusun laporan ini, dan divisi terkait akan mengambil tindakan yang sesuai. Setelah dilakukan perbaikan dan dianggap siap untuk pengujian ulang, tim QA akan melakukan pengujian kembali terhadap *bug* atau kesalahan tersebut. Setelah lulus pengujian QA, langkah selanjutnya adalah menyusun dokumentasi proyek agar dapat diakses oleh anggota tim QA lainnya dan tim *Testing Operation* di masa depan.

Dokumentasi proyek berperan penting dalam menjaga konsistensi dan transparansi dalam proses pengembangan aplikasi. Ini mencakup catatan tentang semua *bug* yang ditemukan, tindakan yang diambil untuk memperbaikinya, serta hasil pengujian QA. Dokumentasi ini juga dapat mencakup informasi tambahan seperti spesifikasi fitur, arsitektur aplikasi, dan panduan pengguna.

Dengan memiliki dokumentasi yang lengkap dan terstruktur, tim QA dan *Testing Operation* dapat dengan mudah mengacu pada informasi yang diperlukan untuk memahami sejarah pengembangan aplikasi dan memastikan bahwa semua *bug* telah diperbaiki dengan benar. Hal ini juga membantu dalam meminimalkan risiko kesalahan dan mempercepat proses pengujian di masa mendatang.

Dalam menjalankan pengujian otomatis, PT. Global Loyalty Indonesia menggunakan aplikasi Appium dengan menggunakan Bahasa Pemrograman Java. Untuk menjalankan aplikasi Alflagift, Android Studio dipakai untuk membuka *Emulator*.

### **3.2.2 Uraian Kerja Magang**

Selama periode magang di PT. Global Loyalty Indonesia dari tanggal 19 Februari 2024 hingga 19 Agustus 2024, adanya keterlibatan dalam berbagai proyek yang sedang berlangsung di perusahaan tersebut. Pekerjaan juga tentunya disesuaikan dengan proyek-proyek yang sedang berjalan, memungkinkannya untuk memperoleh pengalaman praktis yang berharga dalam dunia kerja.

Informasi lengkap mengenai tugas-tugas magang dapat dijelaskan dalam Tabel 3.2. Tabel ini memberikan detail komprehensif mengenai aktivitas yang dilakukan selama periode magang, termasuk proyek-proyek yang ditangani, tanggung jawab yang diemban, dan pencapaian yang telah dicapai selama periode tersebut. Melalui pengalaman magang ini,

harapannya dapat memperoleh pemahaman yang mendalam tentang proses kerja di PT. Global Loyalty Indonesia, serta mengembangkan keterampilan dan pengetahuan yang relevan dengan industri tersebut. Hal ini akan menjadi landasan yang kuat untuk pengembangan karir di masa depan.

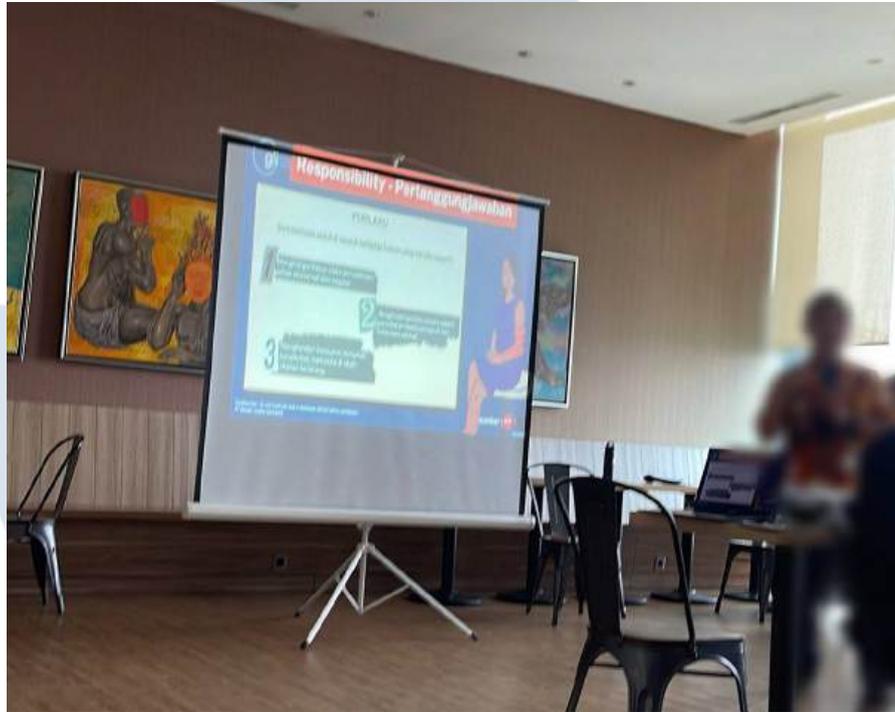
**Tabel 3.2 Uraian Kerja Magang**

Tanggal	Uraian Kerja Magang
19 Februari 2024	<ul style="list-style-type: none"> <li>- Pengenalan perusahaan, <i>briefing</i> dari tim <i>Human Capital</i>,</li> <li>- Pengaturan alat-alat yang akan dipakai dalam proses kerja <i>Quality Assurance (QA)</i>,</li> </ul>
20-24 Februari 2024	Mempelajari <i>query</i> pada <i>database</i> , serta berpartisipasi dalam pelaksanaan <i>Full Cycle Rollout</i> pada aplikasi AlfaGift
26 Februari 2024	Bergabung dengan tim SCRUM
26 Februari - 29 Maret	<i>Project e-Service</i> (pembuatan skenario, <i>testing</i> , <i>bug</i> , dokumentasi, serta laporan)
1 April - 30 April 2024	<i>Project Virtual Product</i> (pembuatan skenario, <i>testing</i> , <i>bug</i> , dokumentasi, serta laporan)

Tabel 3.2 menampilkan gambaran tentang tanggal-tanggal spesifik dan *detail* pekerjaan magang yang dijalankan selama periode magangnya di PT. Global Loyalty Indonesia. Setiap baris tabel menunjukkan tanggal tertentu dan uraian pekerjaan yang dilakukan pada tanggal tersebut.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

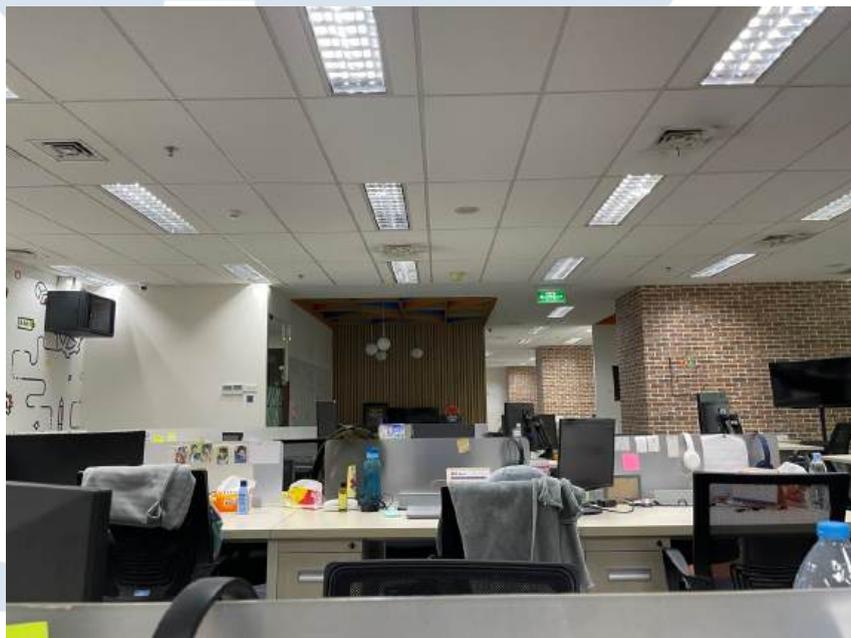
### 3.2.2.1 Pengenalan Perusahaan dan *Tools* yang digunakan



**Gambar 3.5 Pengenalan dan Sosialisasi Magang**

Pada awal hari pertama magang, diawali dengan perkenalan mengenai PT. Global Loyalty Indonesia, yang meliputi pemaparan mengenai pola kerja yang akan dijalani selama program magang. Acara kemudian dilanjutkan dengan penandatanganan kontrak magang dan penyambutan dari CEO PT. Global Loyalty Indonesia. Pada Gambar 3.5 menggambarkan suasana saat acara sosialisasi magang di PT. Global Loyalty Indonesia. Pada hari tersebut, diperkenalkannya dengan budaya perusahaan, aturan mengenai jam kerja, serta prosedur perizinan yang dijelaskan dengan *detail*. Ini adalah langkah awal yang penting bagi para magang untuk memahami lingkungan kerja dan menyesuaikan diri dengan tata tertib serta nilai-nilai perusahaan yang diperlukan untuk berhasil selama program magang.

Para peserta magang di PT. Global Loyalty Indonesia diberi kesempatan untuk mengenal lebih dalam budaya perusahaan serta sesama peserta magang. Sebelum tur keliling kantor dimulai, mereka berkumpul berdasarkan departemen magang masing-masing. Saat tur keliling kantor, yang meliputi dua lantai, diberikan penjelasan tentang berbagai divisi di perusahaan untuk memudahkan komunikasi di antara mereka. Setelah itu, peserta magang diajak ke lantai 29, dimana pada lantai ini, divisi yang tercakup fokus pada pengembangan aplikasi Alfagift seperti tim *mobile developer*, manajer proyek, *product operation*, dan lain lain. Ini adalah pengalaman yang sangat berharga karena dapat lebih memahami struktur organisasi perusahaan dan bagaimana setiap divisi berkontribusi dalam mencapai tujuan bersama.



**Gambar 3.6 Meja divisi QA**

Gambar 3.6 adalah ruangan di lantai 29 yang menjadi barisan divisi QA. Di sana akan diperkenalkan kepada seluruh anggota tim QA. Setelah proses perkenalan, selanjutnya akan diarahkan ke tempat duduk beserta meja yang akan digunakan selama pelaksanaan magang di PT.

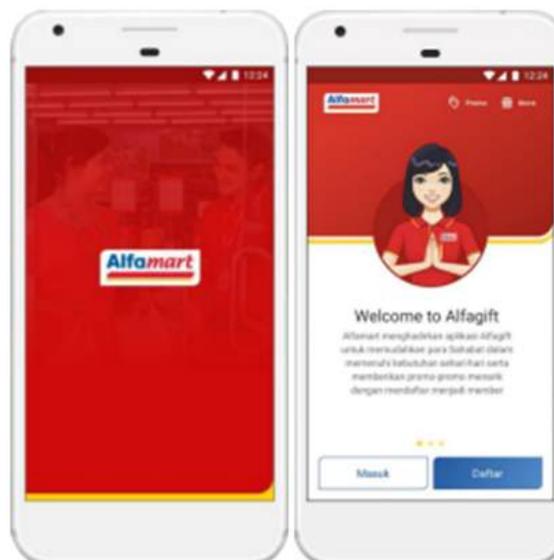
Global Loyalty Indonesia. Sebelum mulai menjalankan tugasnya sebagai QA, tentunya adanya diskusi dengan anggota tim QA lainnya dan mendapat pembagian mentor. Langkah ini bertujuan untuk memastikan pemahaman yang baik terhadap tugas-tugas magang serta memungkinkan dalam berpartisipasi diproyek-proyek yang dipimpin oleh mentor masing-masing. Sehingga proses magang dapat lebih efektif dalam mengintegrasikan ke dalam tim dan menjalankan tanggung jawabnya sebagai QA dengan baik.

### USER GUIDE APLIKASI ALFAGIFT

#### 1. WELCOME SCREEN

Welcome screen Pertama adalah tampilan pertama saat aplikasi ini di buka, berdurasi selama 3 detik dan setelah itu akan tampil. Welcome screen Kedua :

1. Promo: Tap untuk melihat Promo, promo yang ada di aplikasi ini tanpa login atau register
2. Store Locator: Tap untuk melihat toko, toko di lokasi terdekat user
3. Text Slider: Swipe untuk membaca informasi di welcome screen
4. Login or Register: Tap untuk login atau register



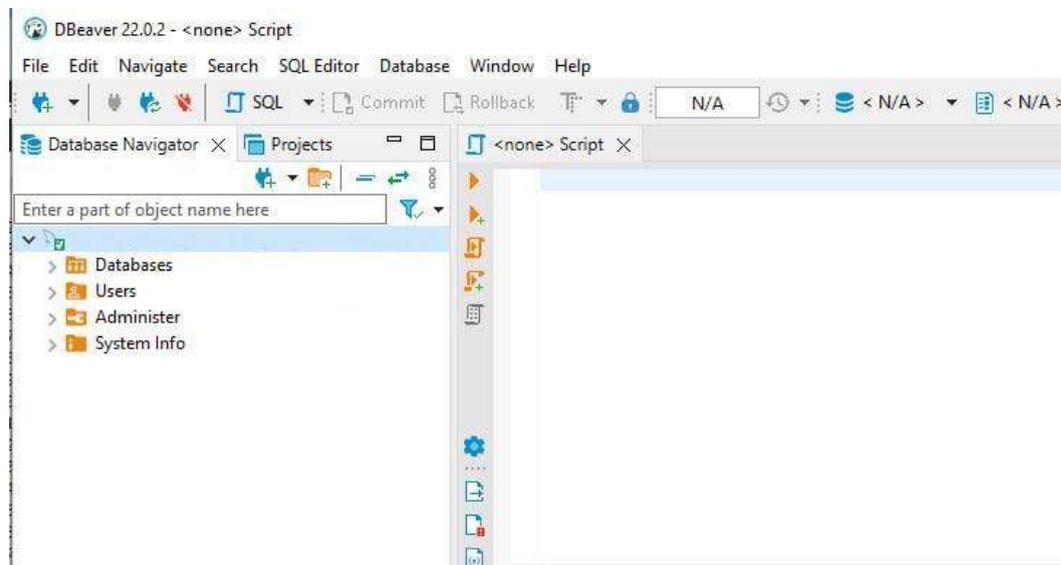
**Gambar 3.7 Contoh Panduan Pengguna aplikasi AlfaGift**

Pada awal bergabung, akan diberi arahan untuk membaca panduan pengguna dari aplikasi Alfagift. Panduan ini mencakup informasi tentang

fungsi dan cara kerja aplikasi, serta penjelasan tombol-tombol pada setiap halaman. Selain itu, panduan tersebut juga menjelaskan konsep *A-Poin*, keanggotaan, promo, dan fitur lainnya. Setelah membaca panduan tersebut akan memahami alur dan fitur aplikasi Alfagift secara menyeluruh. Ini adalah langkah penting untuk memastikan memiliki pemahaman yang kuat tentang cara menggunakan aplikasi serta fitur-fitur yang tersedia sebelum melakukan tugas-tugas berikutnya dalam proyek. Dengan pemahaman yang baik tentang aplikasi, kerja dapat lebih efektif dalam melakukan pengujian, memberikan umpan balik, dan berkontribusi dalam pengembangan aplikasi secara keseluruhan. Keesokan harinya, diberikan tugas langsung melaksanakan *fullcycle rollout* pada aplikasi tersebut dengan menguji semua fitur. Langkah ini diambil setelah mempelajari penjelasan fitur pada hari sebelumnya. Dengan demikian, dapat langsung menerapkan pengetahuan yang baru dipelajari ke dalam praktik dengan melakukan pengujian langsung pada semua fitur aplikasi. Ini adalah langkah yang efektif untuk mengasah pemahaman dan keterampilan dalam menggunakan aplikasi serta memastikan bahwa aplikasi tersebut berfungsi dengan baik sebelum dirilis ke pengguna secara resmi.

Divisi QA di PT. Global Loyalty Indonesia menggunakan berbagai tools untuk menjalankan tugasnya, yaitu DBeaver, Appium, MongoDB, Swagger, *Google Docs*, *Google Sheets*, dan Solr. Berikut ini merupakan beberapa penjelasan terkait aplikasi ataupun *tools* yang sering digunakan dalam proses kerja magang.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

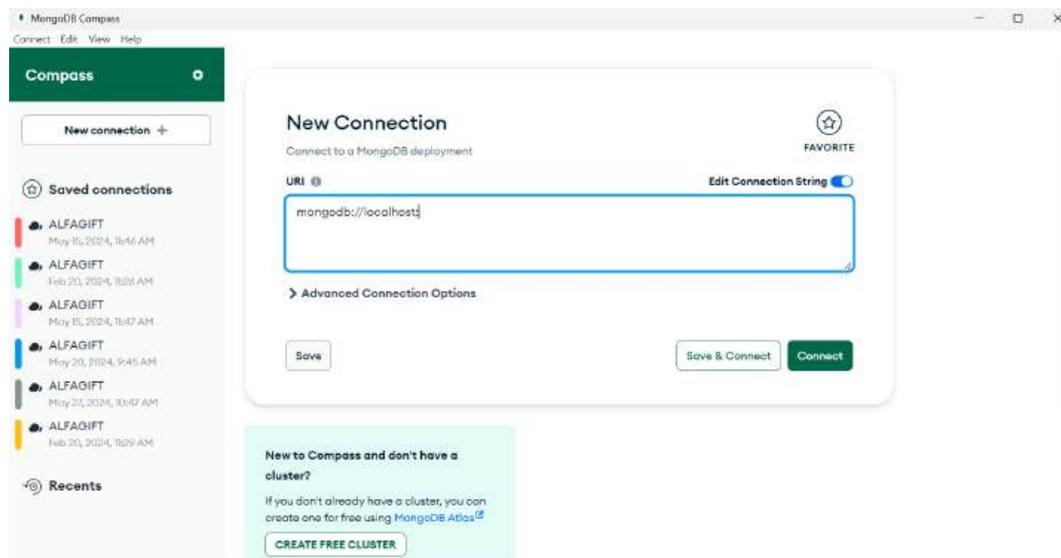


**Gambar 3.8** Tampilan aplikasi DBeaver

Gambar 3.8 menampilkan tampilan aplikasi yang digunakan. Dengan DBeaver, dapat dengan mudah terhubung ke berbagai jenis *database* dan melakukan berbagai tugas administratif dan analitis, seperti menjalankan *query*, memeriksa struktur *database*, mengelola tabel dan indeks, serta mengimpor dan mengekspor data. Aplikasi ini memberikan fleksibilitas dan kemudahan penggunaan dalam menjalankan operasi database yang diperlukan dalam proses pengembangan dan pengujian aplikasi.

Dengan akses yang mudah ke berbagai database dan fungsi yang kaya, DBeaver menjadi salah satu alat yang penting bagi tim QA dalam memeriksa data dan menjalankan *query* yang diperlukan untuk memvalidasi fungsionalitas dan integritas data aplikasi yang sedang diuji.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

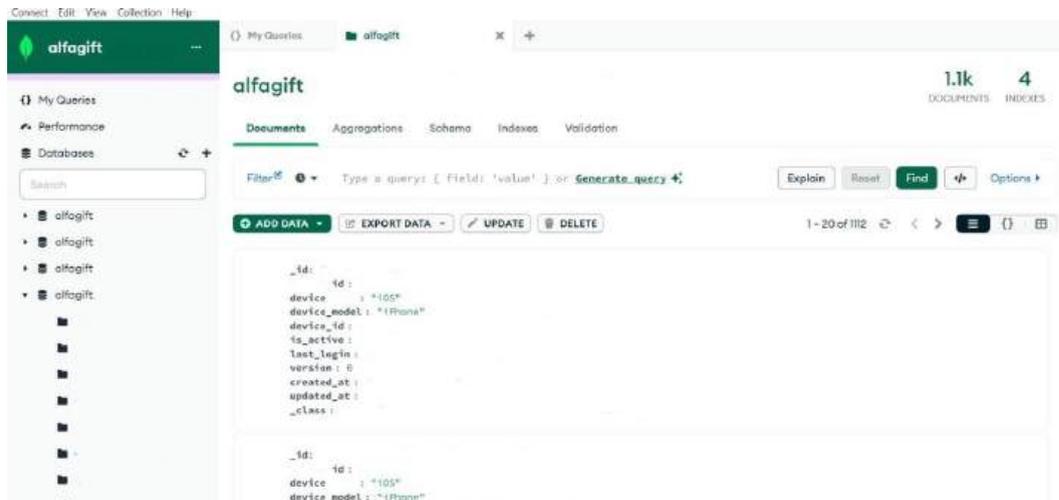


**Gambar 3.9** Tampilan aplikasi MongoDB

Gambar di atas menampilkan tampilan aplikasi MongoDB, yang digunakan untuk penyimpanan *database*. MongoDB adalah salah satu sistem manajemen basis data NoSQL yang populer, yang digunakan untuk menyimpan dan mengelola data dalam format dokumen. Aplikasi ini memberikan antarmuka grafis yang memudahkan pengguna untuk berinteraksi dengan basis data MongoDB.

Dalam MongoDB, data disimpan dalam format dokumen JSON yang fleksibel, yang memungkinkan untuk menyimpan data dengan struktur yang dinamis dan berubah-ubah. Aplikasi ini menyediakan berbagai fitur untuk mengelola basis data, termasuk membuat, mengedit, dan menghapus koleksi data, menjalankan *query*, mengindeks data, serta memantau kinerja basis data.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 3.10 Koneksi MongoDB**

Gambar tersebut merupakan contoh ketika salah satu koneksi database pada MongoDB diklik, yang menghasilkan tampilan data dari database tersebut. Ketika sebuah koneksi database MongoDB diakses, aplikasi MongoDB *Client* akan menampilkan struktur dan konten dari database yang terkait. Ini termasuk daftar koleksi yang ada di dalam database tersebut, serta data yang terkandung di dalam koleksi-koleksi tersebut.

Dengan mengklik koneksi database, pengguna dapat melihat detail dari setiap koleksi yang ada di dalamnya. Ini mencakup struktur dokumen, tipe data, dan nilai-nilai yang tersimpan dalam koleksi tersebut. Pengguna juga dapat menjalankan *query* untuk melakukan pencarian, pengeditan, atau penghapusan data, serta melihat riwayat operasi yang dilakukan terhadap data.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 3.11 Solr**

Gambar di atas menampilkan tampilan dari Solr, yang merupakan salah satu *platform* pencarian dan analisis teks terkemuka yang banyak digunakan dalam pengembangan aplikasi. Solr tidak hanya berfungsi sebagai mesin pencari, tetapi juga sebagai penyimpanan data yang cepat dan *scalable*. Dalam tampilan Solr, pengguna dapat melihat berbagai informasi terkait indeks dan data yang disimpan di dalamnya. Mereka dapat menjelajahi berbagai kunci pencarian, mengatur parameter pencarian, dan menganalisis statistik pencarian untuk memahami pola pencarian pengguna.

Dengan Solr, pengguna, termasuk tim QA, dapat memanfaatkan kemampuan pencarian dan penyimpanan data yang kuat untuk memvalidasi fungsi pencarian dalam aplikasi, serta melakukan analisis terhadap data yang tersimpan di dalamnya. Ini menjadi penting dalam proses pengujian aplikasi yang memanfaatkan fitur pencarian, karena memungkinkan tim QA untuk memverifikasi bahwa operasi pencarian berjalan dengan baik dan memberikan hasil yang relevan dan akurat kepada pengguna.



**Gambar 3.12 Appium**

Dalam menjalankan *automation testing*, PT. Global Loyalty Indonesia mengadopsi Appium sebagai alat utama dengan Bahasa pemrograman Java. Appium dipilih karena kemampuannya yang mendukung pengujian aplikasi lintas *platform* dan memungkinkan pengujian pada perangkat nyata serta emulator. Selain itu, Bahasa pemrograman Java dipilih karena popularitasnya dan kesesuaiannya dengan penggunaan Appium.

Untuk menjalankan aplikasi Alfagift pada pengujian Android, Android Studio digunakan untuk membuka Emulator. Melalui Android Studio, tim QA dapat membuat, menguji, dan *mendebug* aplikasi Android dengan mudah. Emulator Android yang disediakan oleh Android Studio memungkinkan pengujian aplikasi di berbagai konfigurasi perangkat dan versi Android.

Sementara itu, untuk mengelola kode *automation testing*, tim menggunakan IntelliJ IDEA sebagai editor kode. IntelliJ IDEA dipilih karena memiliki berbagai fitur yang mendukung pengembangan dan pengujian, termasuk fitur-fitur khusus yang berguna dalam proses *automation testing*. *Editor* ini memfasilitasi penulisan, pengeditan, dan *debug* kode *automation testing* dengan efisien.



Pada Gambar 3.13, menampilkan skenario yang digunakan sebagai panduan untuk melakukan pengujian aplikasi Alfagift. Pengujian dilakukan pada versi 427005 dari aplikasi tersebut. *Fullcycle Rollout* Aplikasi Alfagift ini dilaksanakan pertama kali pada tanggal 20 Februari 2024. Mayoritas skenario yang diuji selama *Fullcycle Rollout* adalah kasus positif dalam aplikasi, dimana tujuannya adalah memastikan bahwa tidak ada konflik antara fitur-fitur yang ada yang dapat menyebabkan kegagalan aplikasi. Pengerjakan pengujian *Fullcycle Rollout* menggunakan perangkat Android maupun iOS. Hal ini menunjukkan komprehensifnya pendekatan pengujian yang dilakukan untuk memastikan kualitas dan kinerja aplikasi Alfagift di berbagai *platform*.

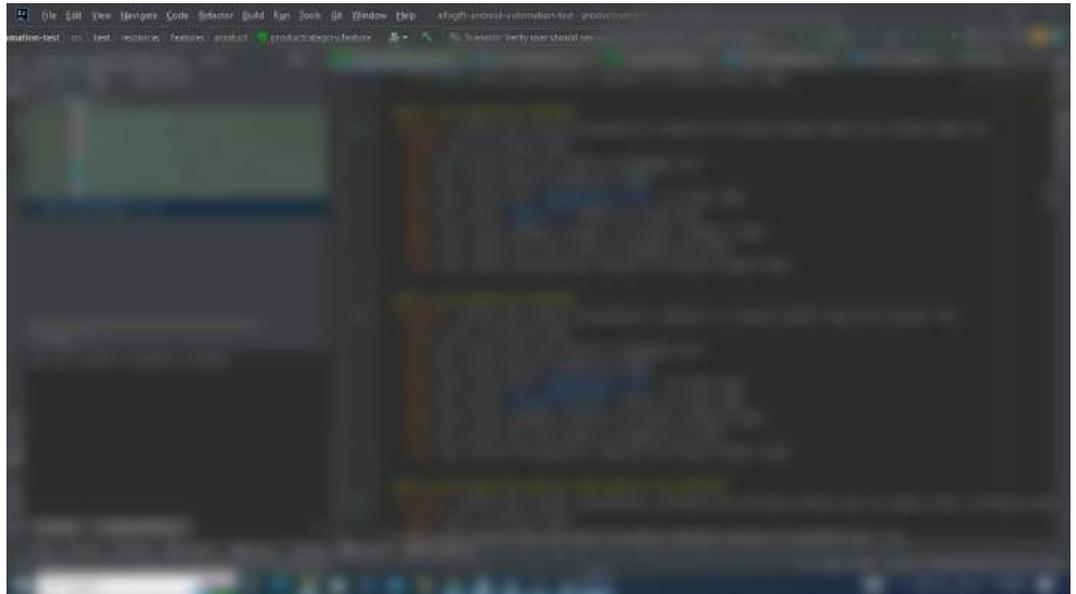
Pelaksanaan *fullcycle rollout* kedua dilaksanakan pada tanggal 14 Mei 2024 dengan versi aplikasi 4280029. Pada kesempatan ini, tim QA bertanggung jawab untuk memastikan bahwa fitur-fitur baru atau perbaikan pada aplikasi Alfagift berjalan dengan baik dan tidak mengganggu fitur lainnya yang sudah berjalan. Dalam pelaksanaan *fullcycle rollout* kedua ini, dengan menggunakan perangkat iOS untuk melakukan pengujian berdasarkan skenario yang telah tersedia. Langkah ini penting untuk memastikan bahwa pengalaman pengguna tetap optimal dan aplikasi beroperasi dengan baik di berbagai *platform* dan perangkat. Dengan demikian, tim QA dapat memastikan kualitas dan keandalan aplikasi sebelum dirilis secara resmi kepada pengguna. Jika menemukan *bug* atau *error* selama melakukan pengujian, langkah pertama yang harus dilakukan adalah mencatatnya. Ini bisa dilakukan dengan membuat catatan yang mencakup deskripsi singkat tentang *bug* atau *error* tersebut, langkah-langkah untuk memperbaiki masalahnya (jika mungkin), serta informasi tambahan yang relevan seperti tangkapan layar atau log kesalahan. Setelah mencatat *bug* atau *error* tersebut, langkah selanjutnya adalah menginformasikan kepada supervisor atau tim yang bertanggung jawab atas pengembangan aplikasi. Ini memungkinkan mereka untuk

mengetahui masalah tersebut dan memprosesnya lebih lanjut, baik dengan memperbaiki langsung *bug* tersebut atau menambahkannya ke dalam backlog untuk diperbaiki di masa mendatang. Dengan cara ini, dapat berperan dalam meningkatkan kualitas aplikasi dengan mengidentifikasi dan melaporkan *bug* atau *error* yang ditemukan selama pengujian.

Pengerjaan *fullcycle rollout* yang berulang-ulang membutuhkan pendekatan yang efisien dan dapat diandalkan untuk memastikan kualitas aplikasi yang konsisten setiap kali dirilis. Oleh karena itu, penggunaan *automation testing* menjadi sangat penting dalam konteks ini.

Dengan menggunakan *automation testing*, tim QA dapat membuat skrip pengujian yang otomatis untuk menjalankan serangkaian tes fungsionalitas dan kinerja pada setiap iterasi *fullcycle rollout*. Skrip ini akan mengotomatiskan proses pengujian yang repetitif dan membebaskan waktu dan sumber daya manusia untuk fokus pada tugas-tugas yang lebih kompleks dan kritis.

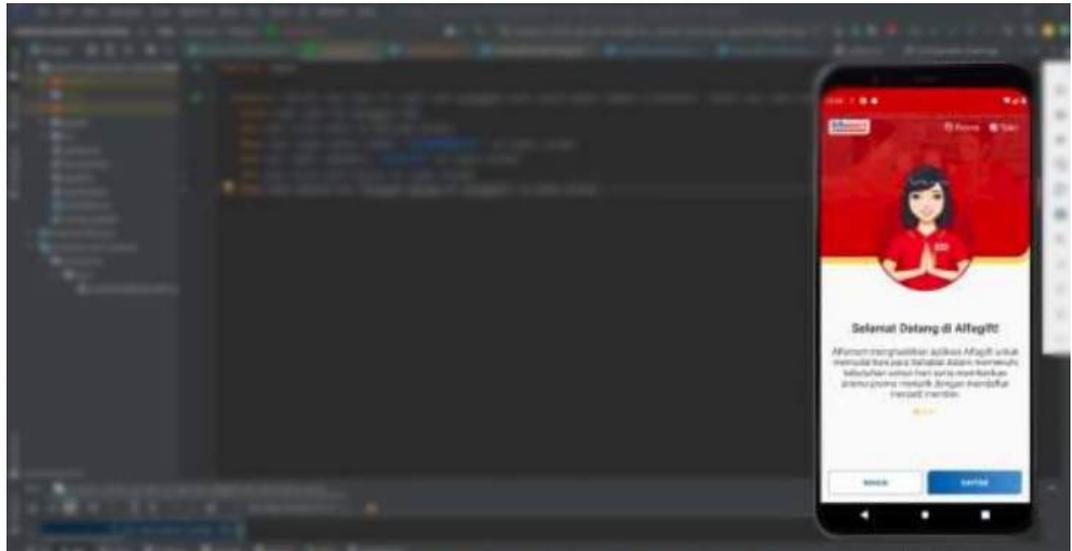
Dalam konteks ini, *automation testing* dijalankan menggunakan aplikasi Appium dengan bahasa pemrograman Java. Appium adalah salah satu *platform* yang populer digunakan untuk melakukan pengujian otomatis pada aplikasi *mobile*, sementara Java adalah bahasa pemrograman yang sering digunakan dalam pengembangan aplikasi dan *testing* karena fleksibilitasnya dan banyaknya dukungan yang tersedia.



**Gambar 3.14 Kode *Automation Testing***

Gambar 3.14 menampilkan tampilan sebagian kode untuk melakukan pengujian pada aplikasi alfagift. Dengan ini, tim QA dapat membuat skrip pengujian yang kuat dan dapat diandalkan untuk memastikan bahwa setiap iterasi *fullcycle rollout* memenuhi standar kualitas yang ditetapkan. Tahapan selanjutnya adalah dengan eksekusi kode tersebut. dengan memastikan emulator beserta server sudah nyala, maka jika berhasil akan menampilkan *welcome screen* dari aplikasi alfagift.

U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



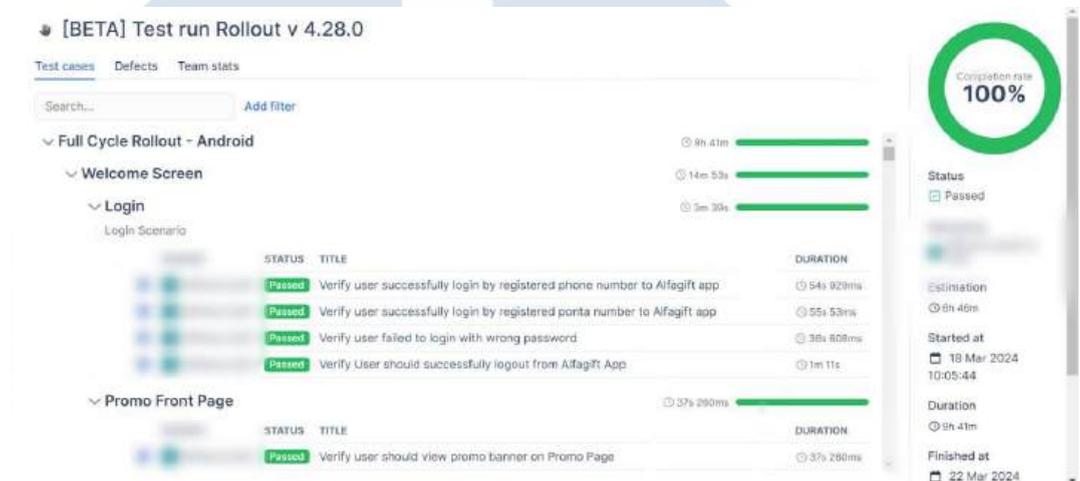
**Gambar 3.15 Tampilan welcome screen**

Gambar 3.15 menampilkan bahwa kode yang dijalankan berhasil menampilkan *welcome page* dari aplikasi Alfigift. Ini menandakan bahwa skrip atau program *automation* telah berhasil membuka aplikasi Alfigift dan mencapai halaman selamat datang (*welcome page*) tanpa mengalami kesalahan.

Dengan berhasilnya langkah ini, pengguna atau tim QA dapat melanjutkan pengujian skenario lanjutan melalui program automation ini. Skenario tersebut mungkin mencakup serangkaian langkah pengujian yang melibatkan navigasi melalui berbagai fitur dan layar aplikasi, mengisi formulir, mengirimkan data, dan memverifikasi hasilnya. Dengan menggunakan *automation*, pengujian dapat dilakukan dengan cepat, konsisten, dan efisien, memastikan bahwa aplikasi Alfigift berjalan dengan baik sesuai dengan ekspektasi sebelum dirilis ke pengguna.

Tahap terakhir dalam proses otomatisasi pengujian adalah pembuatan laporan (*report*) yang akan memberikan ringkasan tentang hasil pengujian kepada anggota tim proyek dan pemangku kepentingan lainnya. Untuk melakukan ini, tim QA dapat menggunakan alat manajemen

pengujian seperti Qase, yang dirancang khusus untuk menghasilkan laporan pengujian yang terperinci dan mudah dibaca.



**Gambar 3.16 Laporan *Automation Testing***

Dengan Qase, tim QA dapat membuat laporan yang mencakup informasi penting seperti hasil pengujian, metrik kinerja, statistik kesalahan, dan rekomendasi perbaikan. Laporan ini dapat disesuaikan dengan kebutuhan tim dan dapat berisi grafik, tabel, dan visualisasi lainnya untuk memudahkan pemahaman seperti pada gambar 3.16.

Melalui laporan yang dihasilkan oleh Qase, anggota tim proyek dan pemangku kepentingan lainnya dapat dengan cepat memahami status pengujian, masalah yang ditemukan, dan langkah-langkah yang harus diambil selanjutnya. Ini membantu dalam pengambilan keputusan yang tepat dan memastikan bahwa aplikasi siap untuk dirilis ke pengguna akhir.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.2.2.3 SCRUM

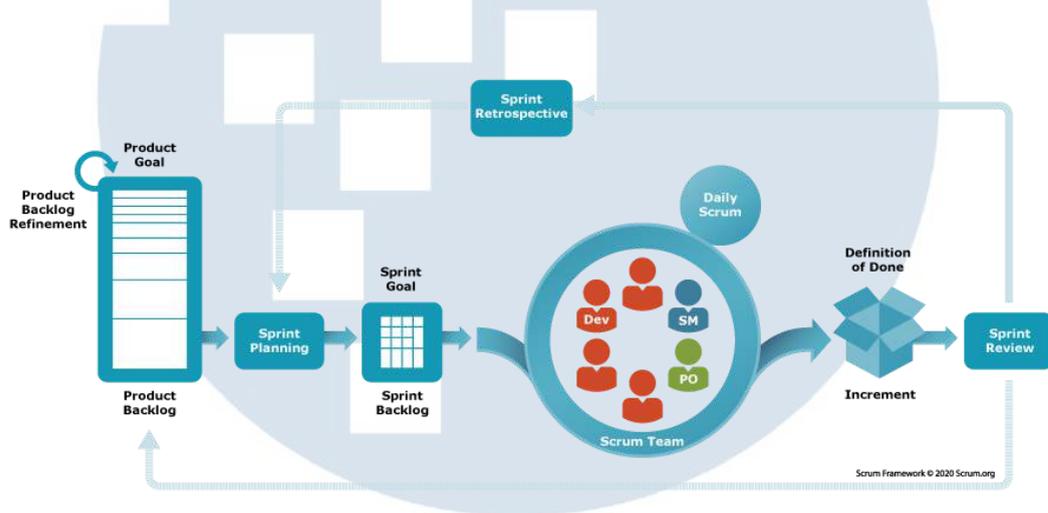
Scrum adalah sebuah kerangka kerja pengembangan perangkat lunak yang berbasis pada pendekatan *Agile*. Pendekatan ini menekankan pada kerja kolaboratif, fleksibilitas, dan iterasi cepat dalam menghasilkan produk yang bernilai. Dalam Scrum, pengembangan dilakukan dalam iterasi kecil yang disebut sprint, yang biasanya berlangsung antara satu hingga empat minggu. Setiap sprint dimulai dengan perencanaan, dilanjutkan dengan pengerjaan, dan diakhiri dengan review dan retrospektif untuk memperbaiki proses kedepannya.

Sebagai seorang QA (*Quality Assurance*) di tim Scrum, peran QA sangat penting dalam memastikan bahwa produk yang dihasilkan memenuhi standar kualitas yang ditetapkan. Berikut ini adalah beberapa peran yang biasanya dimainkan oleh seorang QA dalam tim Scrum:

1. Mengembangkan dan Mengelola Kasus Uji (*Test Cases*): Bertanggung jawab untuk merencanakan, membuat, dan mengelola kasus uji untuk memastikan bahwa semua fitur dan fungsionalitas yang dikembangkan telah diuji secara menyeluruh.
2. Pelaksanaan Pengujian (*Testing*): Melakukan pengujian perangkat lunak, baik secara manual maupun otomatis, untuk memverifikasi bahwa fitur-fitur yang dikembangkan bekerja sesuai yang diharapkan dan memenuhi kebutuhan pengguna.
3. Partisipasi dalam *Review Sprint* dan *Demo*: Terlibat dalam *review sprint* dan *demo* untuk memberikan masukan tentang kualitas produk yang dikembangkan dan apakah ada perubahan atau perbaikan yang perlu dilakukan.
4. Kolaborasi dengan Tim Pengembangan: Bekerja sama dengan tim pengembangan untuk memastikan bahwa masalah-masalah yang ditemukan selama pengujian diperbaiki dengan cepat dan efisien.
5. Membantu dalam Meningkatkan Proses: Membantu dalam retrospektif *sprint* untuk mengidentifikasi area di mana proses pengembangan dan

pengujian dapat ditingkatkan, serta memberikan masukan untuk memperbaiki proses tersebut.

6. Memastikan Kualitas Produk Secara Keseluruhan: Sebagai bagian dari tim, Anda memiliki tanggung jawab untuk memastikan bahwa produk yang dihasilkan memiliki tingkat kualitas yang tinggi dan memenuhi harapan pengguna.



**Gambar 3.17 Kerangka Kerja Scrum**

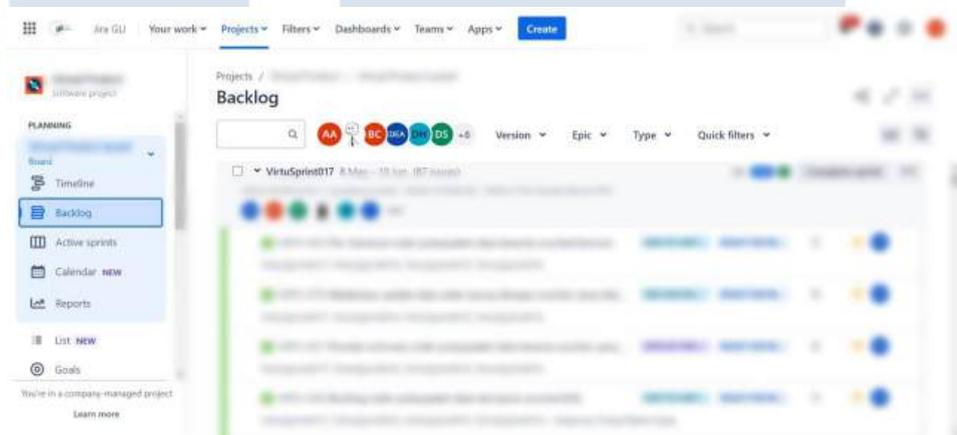
Gambar 3.17 merupakan representasi visual dari kerangka kerja Scrum, yang terdiri dari beberapa elemen utama yang digunakan dalam pengembangan perangkat lunak berbasis Scrum.

### 3.2.2.3.1 *Product Backlog*

*Product backlog* dalam kerangka kerja Scrum adalah daftar prioritas yang berisi semua pekerjaan yang perlu dilakukan untuk mengembangkan produk secara keseluruhan. Ini merupakan salah satu dari tiga artefak utama dalam Scrum. *Product backlog* dibuat dan dikelola oleh *Product Owner*, pada tim SCRUM saat ini yaitu Pak Budi yang bertanggung jawab untuk memastikan bahwa daftar ini selalu diperbarui dan menggambarkan

kebutuhan dan keinginan para pemangku kepentingan (stakeholders) produk. Produk *backlog* mencakup semua fitur, perbaikan, perubahan, dan pekerjaan lain yang perlu dilakukan untuk meningkatkan produk.

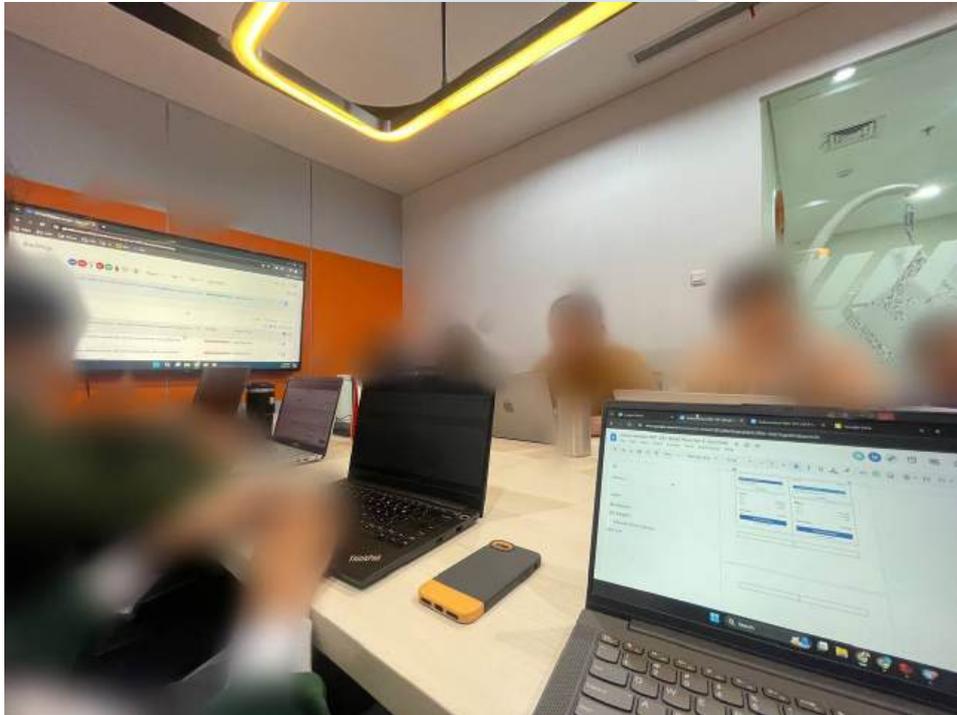
Prioritas dalam produk backlog ditetapkan berdasarkan nilai bisnis yang diharapkan dan kebutuhan pengguna. *Item-item* yang memiliki nilai lebih tinggi atau memenuhi kebutuhan kritis biasanya ditempatkan di bagian atas *backlog*, sedangkan *item-item* yang kurang penting ditempatkan di bagian bawah.



**Gambar 3.18 Backlog**

Gambar di atas adalah contoh *backlog* yang digunakan oleh tim. *Backlog* ini merupakan daftar dari semua *item* pekerjaan yang perlu dilakukan dalam proyek. *Item-item* ini bisa berupa fitur baru yang perlu ditambahkan, perbaikan yang harus dilakukan, atau tugas-tugas lain yang diperlukan untuk mengembangkan produk. *Product Backlog* disusun dalam urutan prioritas, dengan *item* yang paling penting atau yang paling bernilai bagi pengguna berada di bagian atas daftar. Dengan menyusun *backlog* ini secara terstruktur dan berdasarkan prioritas, tim dapat fokus pada pekerjaan yang paling penting dan memastikan bahwa proyek berjalan sesuai dengan kebutuhan dan tujuan yang ditetapkan.

### 3.2.2.3.2 *Sprint Planning*



**Gambar 3.19 *Sprint Planning***

*Sprint Planning* adalah pertemuan yang terjadi di awal setiap sprint. Tim menggunakan waktu ini untuk memilih *item-item* dari *Product Backlog* yang akan dikerjakan selama sprint tersebut. *Sprint planning* juga digunakan untuk menentukan bagaimana tim akan menyelesaikan pekerjaan tersebut dan menetapkan tujuan sprint.

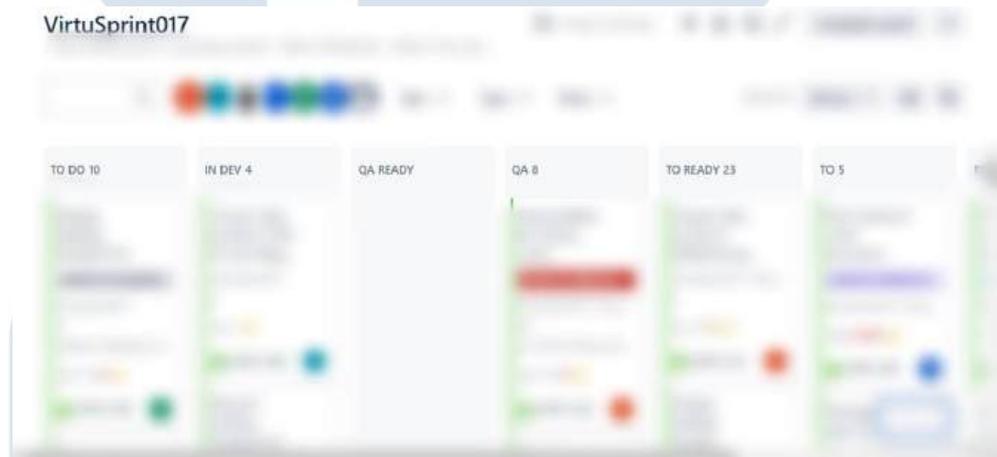
Pertemuan ini terdiri dari dua bagian. Pertama, tim pengembangan dan *Product Owner* (pemilik produk) berkolaborasi untuk memilih *item* dari *Product Backlog* yang akan dimasukkan ke dalam *Sprint Backlog* untuk dikerjakan selama *Sprint* ini. Mereka mempertimbangkan prioritas, kebutuhan pengguna, dan kemampuan tim dalam menyelesaikan pekerjaan.

Kemudian, dalam bagian kedua, tim pengembangan merencanakan bagaimana mereka akan menyelesaikan pekerjaan tersebut. Mereka

membahas strategi, memperkirakan waktu yang diperlukan untuk setiap tugas, dan membagi pekerjaan sesuai dengan kemampuan dan keahlian anggota tim.

Tujuan *Sprint Planning* adalah untuk menghasilkan rencana yang jelas dan terstruktur tentang apa yang akan dilakukan selama *Sprint*, sehingga tim memiliki pemahaman yang sama tentang tujuan dan tugas yang harus diselesaikan. Ini membantu memastikan bahwa tim fokus pada pekerjaan yang paling penting dan dapat bekerja secara efisien selama *Sprint*.

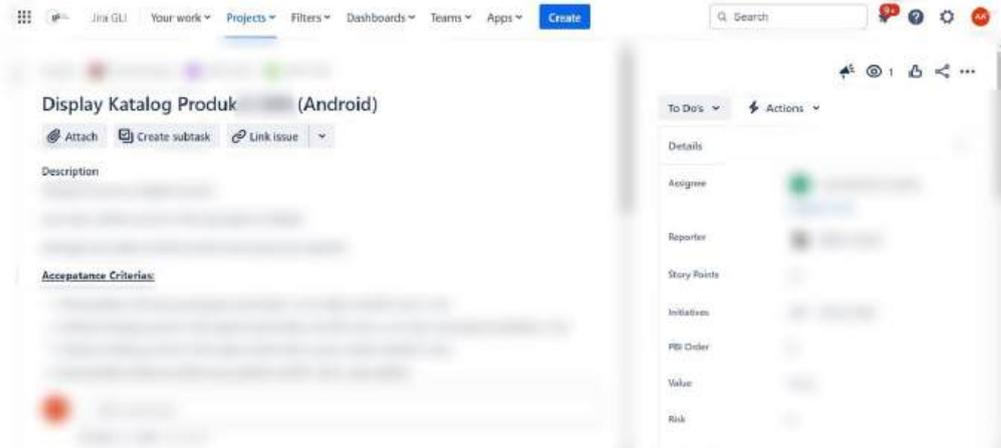
### 3.2.2.3.3 *Sprint Backlog*



**Gambar 3.20 *Sprint Backlog***

*Sprint backlog* merupakan daftar tugas yang tim pilih dengan cermat selama rapat perencanaan sprint. Daftar ini berisi detail mengenai pekerjaan yang harus diselesaikan dalam periode sprint tersebut. Di sini, tim mencantumkan siapa yang bertanggung jawab atas setiap tugas serta perkiraan waktu yang diperlukan untuk menyelesaikannya. Gambar 3.20 adalah representasi visual dari *sprint backlog* yang menjadi panduan utama bagi tim dalam menjalankan proyek. Setiap kali sebuah tugas

selesai, tim akan memindahkannya ke tahap berikutnya, yang mungkin adalah proses pengujian. Ini membantu menjaga alur kerja tetap teratur dan memastikan bahwa setiap langkah dalam proyek berjalan sesuai rencana.



**Gambar 3.21 Detail Sprint Backlog**

Gambar 3.21 menampilkan salah satu bagian dari *sprint backlog* yang menampilkan detail tentang pekerjaan yang harus dilakukan, siapa yang bertanggung jawab atas setiap tugas, dan perkiraan waktu yang dibutuhkan untuk menyelesaikan setiap tugas tersebut. Dengan informasi ini, anggota tim dapat dengan jelas memahami tanggung jawab mereka dan mengatur waktu mereka sesuai dengan target yang telah ditetapkan. Hal ini membantu menjaga kelancaran dan efisiensi dalam menjalankan sprint dan proyek secara keseluruhan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

#### 3.2.2.3.4 Daily Scrum



**Gambar 3.22 Daily SCRUM**

*Daily Scrum* adalah pertemuan harian yang dilakukan oleh tim pengembangan dalam kerangka kerja Scrum. Ini adalah pertemuan singkat yang biasanya berlangsung sekitar 15 menit setiap hari kerja, di mana anggota tim berbagi informasi tentang apa yang mereka kerjakan, apa yang akan mereka kerjakan selanjutnya, dan mengidentifikasi hambatan atau masalah yang menghalangi kemajuan. Di tim Scrum, kami menjalankan daily setiap pagi pukul 9:45. Rutinitas ini telah menjadi bagian penting dari pola kerja kami. Dengan mengatur waktu pertemuan seperti itu, kami memastikan bahwa setiap anggota tim dapat bergabung secara teratur untuk berbagi pembaruan, merencanakan tindakan, dan menyinkronkan langkah-langkah berikutnya. Dengan menjalankan daily standup pada jam yang sama setiap hari, kami membangun kebiasaan yang konsisten dan terorganisir. Ini membantu dalam menciptakan suasana kerja yang terstruktur dan produktif. Kami dapat dengan mudah menyinkronkan

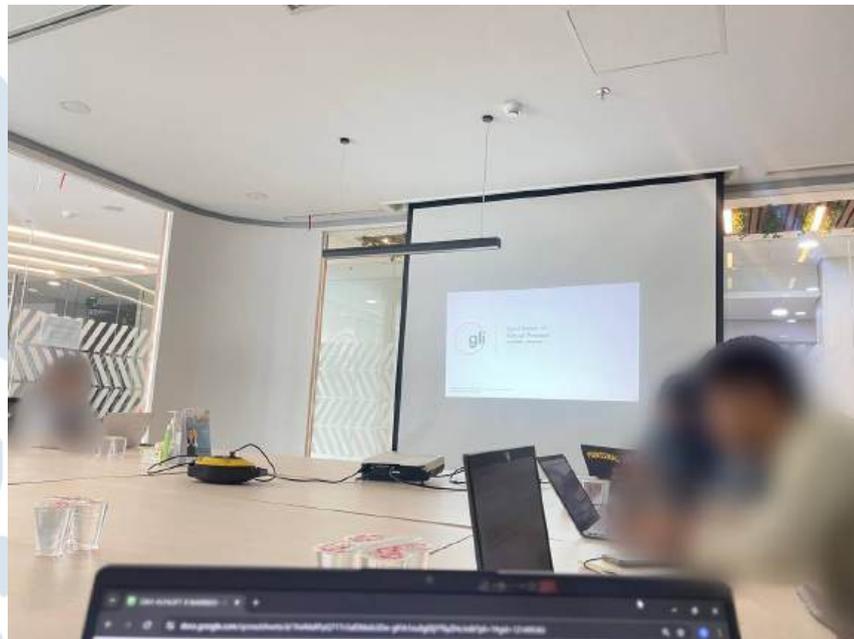
jadwal kami, menjadikan pertemuan ini sebagai titik fokus untuk memulai hari dengan langkah yang jelas dan terkoordinasi.

Selama *Daily Scrum*, setiap anggota tim menjawab tiga pertanyaan sederhana:

1. Apa yang dikerjakan sejak pertemuan terakhir?
2. Apa yang akan dikerjakan hingga pertemuan berikutnya?
3. Apakah ada hambatan atau masalah yang menghalangi?

Tujuan dari *Daily Scrum* adalah untuk menjaga tim tetap terinformasi tentang kemajuan proyek, memfasilitasi kolaborasi dan koordinasi antar anggota tim, serta mengidentifikasi masalah secepat mungkin agar dapat diatasi secara efektif. Ini membantu memastikan bahwa tim tetap fokus pada tujuan Sprint dan dapat mengatasi perubahan atau tantangan yang muncul selama pengembangan produk.

#### **3.2.2.3.5 *Sprint Review***



**Gambar 3.23 *Sprint Review***

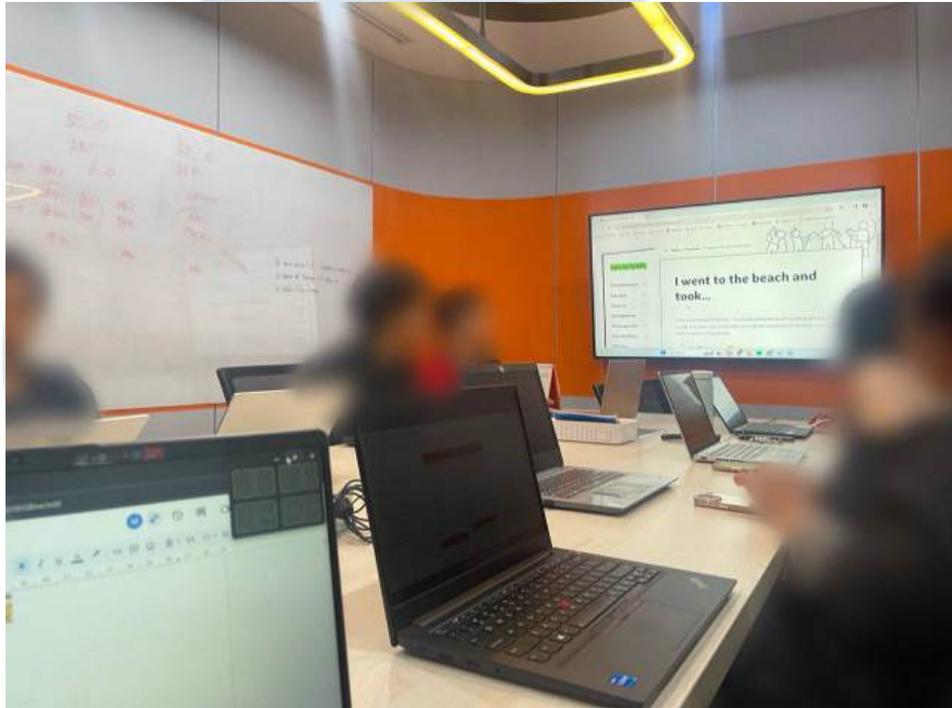
*Sprint review* adalah pertemuan yang diadakan di akhir setiap *Sprint* dalam kerangka kerja Scrum. Ini adalah waktu ketika tim pengembangan menunjukkan apa yang telah mereka kerjakan selama *Sprint* kepada para pemangku kepentingan (*stakeholders*) produk, seperti manajemen produk, pemilik bisnis, atau pengguna akhir. Tujuan utama dari *sprint review* adalah untuk mendapatkan *feedback* dari para pemangku kepentingan tentang pekerjaan yang telah selesai, serta untuk mengevaluasi apakah produk yang dihasilkan memenuhi harapan dan kebutuhan mereka. Tim pengembangan akan memperlihatkan fitur-fitur baru, perubahan, atau peningkatan yang telah mereka lakukan selama *Sprint* kepada para pemangku kepentingan.

*Sprint review* juga merupakan kesempatan bagi tim untuk meninjau produk yang telah mereka kembangkan dan membandingkannya dengan visi produk dan tujuan bisnis yang telah ditetapkan sebelumnya. Hasil dari *sprint review* dapat berupa umpan balik, masukan, atau perubahan ke arah mana produk harus dikembangkan selanjutnya.

Jadi, secara singkat, *sprint review* adalah pertemuan di mana tim pengembangan memperlihatkan apa yang telah mereka kerjakan kepada para pemangku kepentingan produk dan mendapatkan umpan balik tentang hasil kerja mereka..

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.2.2.3.6 *Sprint Retrospective*

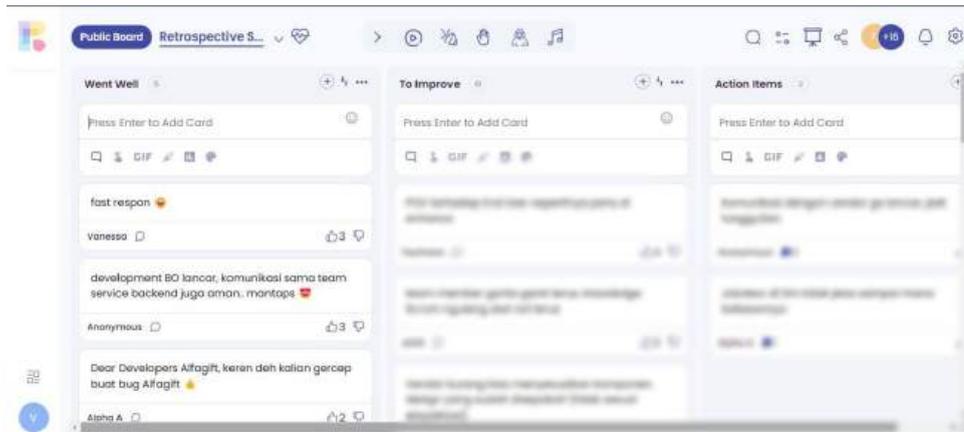


**Gambar 3.24 *Sprint Retrospective***

*Sprint Retrospective* adalah pertemuan yang diadakan di akhir setiap Sprint dalam kerangka kerja Scrum. Ini adalah waktu di mana tim pengembangan berkumpul untuk merefleksikan proses kerja mereka selama *Sprint* yang baru berakhir.

Tujuan utama *Sprint Retrospective* adalah untuk belajar dari pengalaman yang telah dilalui oleh tim selama *Sprint* tersebut. Tim membahas apa yang telah berhasil, apa yang tidak berhasil, dan bagaimana mereka bisa menjadi lebih baik di masa depan.

Dalam *Sprint Retrospective*, tim biasanya melakukan dua hal: mengidentifikasi hal-hal yang telah berjalan baik (positif) dan hal-hal yang perlu diperbaiki (perbaikan). Mereka berbagi pengalaman, memberikan umpan balik, dan mencari solusi untuk meningkatkan proses kerja mereka ke depannya.



**Gambar 3.25 Tools Reetro**

Pada gambar 3.25, terlihat *Sprint Retrospective* yang dilakukan menggunakan alat bantu Reetro. Reetro merupakan salah satu tools yang memungkinkan tim untuk secara langsung memasukkan masukan-masukan yang dapat meningkatkan proses kerja mereka. Dengan menggunakan Reetro, tim dapat dengan mudah berkolaborasi dalam mengevaluasi sprint yang telah berlalu dan menemukan area-area yang perlu diperbaiki atau ditingkatkan. Melalui platform ini, anggota tim dapat memberikan masukan secara terstruktur, memfasilitasi diskusi, dan merencanakan langkah-langkah perbaikan untuk meningkatkan produktivitas dan efisiensi kerja di masa mendatang.

Pentingnya *Sprint Retrospective* adalah untuk menciptakan lingkungan di mana tim merasa aman untuk berbagi pemikiran dan ide, serta untuk merayakan keberhasilan dan mengatasi tantangan bersama-sama. Dengan melakukan ini, tim dapat terus berkembang dan meningkatkan kinerja mereka dari *Sprint* ke *Sprint*.

#### **3.2.2.4 Project e-Service**

Dalam proyek *e-service*, QA memiliki peran penting dalam memastikan kualitas dan kepuasan pengguna. QA bertanggung jawab

untuk mengembangkan prosedur pengujian yang komprehensif, melakukan pengujian fungsional dan non-fungsional, serta memantau kualitas produk selama siklus pengembangan. Selain itu, QA juga bertugas mengidentifikasi *bug*, melaporkannya kepada tim pengembangan, dan memastikan perbaikan yang tepat. Mereka juga terlibat dalam uji penerimaan pengguna untuk memastikan bahwa layanan *e-service* memenuhi harapan pengguna akhir. Dengan keterlibatan mereka yang baik, QA membantu memastikan bahwa layanan *e-service* yang dikembangkan memenuhi standar kualitas yang tinggi dan memberikan nilai yang diharapkan kepada pengguna.



The image shows a screenshot of a testing scenario table. The table has several columns, including 'No.', 'Kasus', 'Langkah', 'Aksi', and 'Kondisi'. The content is mostly blurred, but a few rows are visible. One row is highlighted in yellow, and another in red. The table appears to be a structured list of test cases or scenarios for an e-service application.

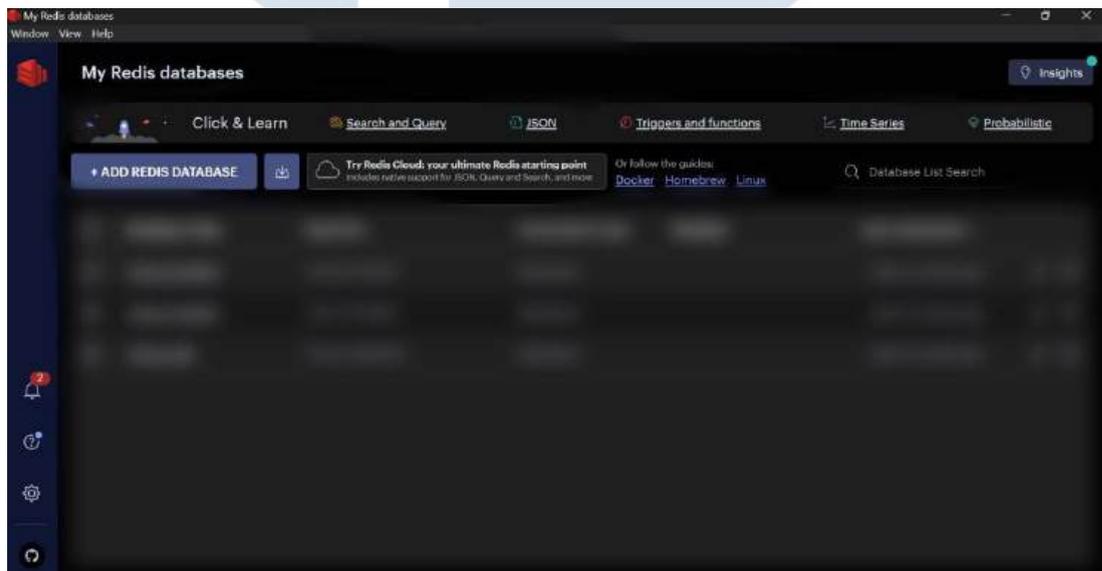
**Gambar 3.26** Skenario *testing e-service*

Pada gambar 3.26, terdapat skenario yang akan digunakan sebagai dasar untuk pengujian kasus. Setelah menyelesaikan penulisan skenario, kemudian memulai proses pengujian menggunakan perangkat Android maupun iOS.

Dalam konteks ini, skenario yang disusun mungkin berisi rangkaian langkah-langkah atau aksi yang harus dijalankan atau diuji oleh pengguna untuk memverifikasi fungsi atau fitur tertentu dari aplikasi atau sistem. Skenario ini dapat mencakup berbagai situasi penggunaan yang mungkin terjadi dalam aplikasi, termasuk pengujian berbagai fitur,

navigasi antarmuka pengguna, dan respons sistem terhadap input pengguna.

Setelah skenario telah disiapkan, kemudian dengan menggunakan perangkat Android dan iOS untuk melakukan pengujian praktis. Ini melibatkan menjalankan skenario yang telah ditetapkan pada perangkat tersebut, mengamati respons aplikasi, dan memverifikasi bahwa fitur atau fungsi berjalan sesuai dengan yang diharapkan. Dengan melakukan pengujian menggunakan berbagai perangkat dan *platform*, seperti Android dan iOS, sehingga dapat memastikan bahwa aplikasi bekerja dengan baik di berbagai lingkungan dan perangkat pengguna, sehingga memastikan pengalaman pengguna yang konsisten dan baik. Ini merupakan langkah penting dalam memastikan kualitas dan keandalan aplikasi sebelum dirilis ke pengguna akhir.



**Gambar 3.27 Tampilan Redis**

Gambar 3.27 menampilkan tampilan aplikasi RedisInsight yang digunakan dalam *project e-service*. Aplikasi ini berfungsi untuk mengedit, mengelola, dan mengatur basis data Redis dengan antarmuka grafis yang intuitif. RedisInsight memberikan pengguna kemampuan untuk

menjelajahi dan memanipulasi data di dalam basis data Redis dengan mudah.

Dengan RedisInsight, pengguna dapat melakukan berbagai operasi, termasuk menambah, mengubah, dan menghapus kunci dan nilai di dalam basis data Redis. Mereka juga dapat melakukan *query* untuk mengekstrak data yang spesifik, serta melihat statistik kinerja dan penggunaan memori dari server Redis.

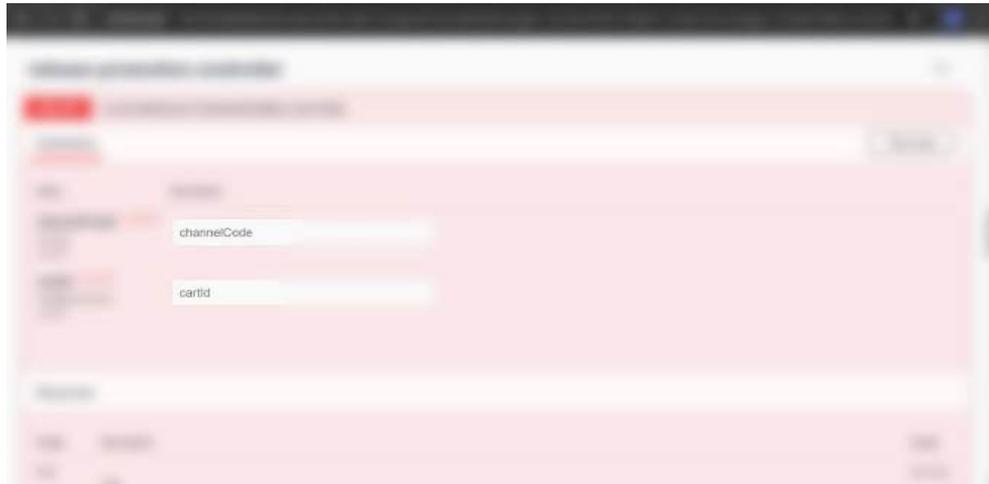
. Dalam proses pengujian proyek *e-service* ini, diharuskan beberapa kali melakukan pelepasan kode *eservice* agar bisa digunakan untuk *testing* secara berulang kali. Maka dari itu agar tidak mengalami kesusahan secara mengulang dalam melepaskan kode tersebut, dilakukannya pembuatan program automation untuk pelepasan kode *eservice*.

#### **3.2.2.4.1 Automation Pelepasan Kode**

Dalam proses pengujian proyek *e-service*, diperlukannya melakukan pelepasan kode *e-service* secara berulang kali agar dapat menguji aplikasi secara menyeluruh. Namun, kesulitan yang mungkin timbul akibat proses pelepasan kode yang berulang tersebut mendorong untuk mencari solusi yang lebih efisien. Oleh karena itu, diputuskan untuk membuat program otomatisasi untuk pelepasan kode *e-service*.

Sebelum membuat otomatisasi pelepasan kode. Diperlukannya membuka website swagger untuk melaksanakan pelepasan kode ini.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 3.28 Website swagger**

Gambar 3.28 menampilkan antarmuka pengguna dari *website* Swagger, di mana harus memasukkan kode satu per satu untuk pelepasan. Kode atau *cartid* ini diperoleh dari hasil skrip yang dijalankan pada aplikasi DBeaver. Dalam konteks ini, digunakannya Swagger untuk melakukan pelepasan kode, yang mengharuskan mereka memasukkan kode secara manual ke dalam antarmuka pengguna Swagger. Proses ini melibatkan penyalinan kode atau *cartid* dari hasil skrip yang dijalankan pada aplikasi DBeaver, dan kemudian memasukkannya secara manual ke dalam antarmuka Swagger. Ini mungkin merupakan langkah yang memakan waktu dan berulang, tetapi diperlukan untuk memastikan bahwa kode yang akan dilepas sesuai dengan yang diinginkan dan sesuai dengan kebutuhan proyek. Dengan demikian, antarmuka Swagger digunakan sebagai alat untuk mengelola proses pelepasan kode dengan memastikan bahwa setiap kode yang dilepas telah ditentukan dengan tepat sesuai dengan kebutuhan aplikasi.

The image shows a screenshot of the DBeaver database management tool. A table named 'tbl\_cart\_id' is displayed, containing a list of numerical values. The table has 8 rows and 1 column. The values are: 46.710, 47.978, 47.641, 47.703, 47.983, 47.809, and 47.583. The table is highlighted in blue.

id	CartId
1	46.710
2	47.978
3	47.641
4	47.703
5	47.983
6	47.809
7	47.583
8	

**Gambar 3.29 CartId pada aplikasi DBeaver**

Gambar 3.29 menampilkan hasil dari sebuah skrip yang menampilkan *cartid* yang akan digunakan dalam website Swagger. Skrip ini mungkin dijalankan di aplikasi seperti DBeaver untuk mengambil data atau informasi tertentu, dalam hal ini *cartid*, yang akan digunakan dalam proses pelepasan kode melalui antarmuka pengguna Swagger. Dengan melihat hasil skrip ini, pengguna, dapat memastikan bahwa mereka memiliki daftar *cartid* yang tepat sebelum melakukan proses pelepasan kode di *website* Swagger. Ini memungkinkan mereka untuk memverifikasi dan memastikan bahwa mereka memasukkan kode yang sesuai dan diperlukan untuk pelepasan kode yang diinginkan. Dengan demikian, gambar 3.29 memberikan pandangan tentang data yang diperoleh dari skrip dan akan digunakan dalam proses selanjutnya dalam pengembangan proyek e-service.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



**Gambar 3.30 Code Pandas**

Gambar 3.30 menampilkan potongan *coding* menggunakan *library* pandas dalam bahasa pemrograman Python yang akan digunakan untuk melaksanakan otomatisasi dalam proses pengembangan. Dalam potongan kode tersebut, pandas kemungkinan digunakan untuk membaca, memanipulasi, dan mengelola data yang diperlukan dalam proses otomatisasi.

Otomatisasi pelepasan kode ini bertujuan untuk meningkatkan efisiensi dan konsistensi dalam proses pengujian. Dengan menggunakan otomatisasi, dapat menghilangkan kebutuhan untuk melakukan pelepasan secara manual setiap kali pengujian dilakukan, yang dapat menghemat waktu dan upaya. Salah satu alat yang digunakan untuk membuat otomatisasi adalah bahasa Python pada aplikasi Visual Studio.

Dengan menggunakan bahasa Python, khususnya dengan *library* seperti Pandas, sehingga dapat membuat skrip otomatisasi yang dapat melakukan serangkaian tugas, mulai dari mengambil kode dari repositori version control, membangun kode, menjalankan tes otomatis, hingga menyebarkan kode ke lingkungan pengujian. Skrip ini dapat dijalankan secara otomatis setiap kali ada perubahan kode di repositori, sehingga tidak perlu lagi melakukan pelepasan kode secara manual.

Melalui pendekatan ini, dapat meningkatkan efisiensi pengujian proyek *e-service*, dengan memastikan bahwa setiap perubahan kode dapat diuji secara menyeluruh tanpa perlu meluangkan waktu untuk proses pelepasan yang manual. Ini juga membantu dalam menjaga konsistensi dalam proses pengujian dan memungkinkan untuk fokus pada pengembangan aplikasi tanpa terganggu oleh tugas-tugas administratif yang berulang. Dengan demikian, pembuatan otomatisasi pelepasan kode menggunakan bahasa Python pada aplikasi Visual Studio menjadi langkah yang cerdas dan efektif bagi tim pengembangan proyek *e-service*.

#### **3.2.2.5 Project Produk Virtual**

Projek Virtual Produk adalah kolaborasi antara Alfagift dengan tim eksternal yang bertujuan untuk menyediakan layanan berbagai fitur produk virtual kepada pengguna. Tujuan utama proyek ini adalah memberikan kemudahan akses dan kenyamanan bagi pengguna, sehingga mereka dapat dengan mudah mengakses layanan-layanan tersebut melalui aplikasi Alfagift. Fitur-fitur ini mencakup berbagai layanan sehingga dengan adanya fitur-fitur ini, pengguna akan memiliki akses yang lebih mudah dan cepat untuk memenuhi kebutuhan mereka secara *digital*, tanpa perlu meninggalkan aplikasi Alfagift.

Scenario ID	Category	Description	Priority	Status	Assigned To
1	Authentication	User login with valid credentials	High	Pass	John Doe
2	Authentication	User login with invalid credentials	High	Pass	John Doe
3	Authentication	User login with expired password	High	Pass	John Doe
4	Authentication	User login with account locked	High	Pass	John Doe
5	Authentication	User login with account disabled	High	Pass	John Doe
6	Authentication	User login with account suspended	High	Pass	John Doe
7	Authentication	User login with account deleted	High	Pass	John Doe
8	Authentication	User login with account not found	High	Pass	John Doe
9	Authentication	User login with account not active	High	Pass	John Doe
10	Authentication	User login with account not verified	High	Pass	John Doe
11	Authentication	User login with account not confirmed	High	Pass	John Doe
12	Authentication	User login with account not approved	High	Pass	John Doe
13	Authentication	User login with account not authorized	High	Pass	John Doe

**Gambar 3.21 Skenario testing produk virtual**

Gambar di atas menampilkan skenario yang akan digunakan untuk pengujian kasus dalam proyek Virtual Produk ini. Langkah awal dalam melaksanakan proyek ini adalah membuat skenario sesuai dengan desain sistem yang telah ditetapkan sebelumnya. Terdapat total 337 baris skenario yang terbagi ke dalam 13 kategori yang berbeda.

Skenario-skenario ini mencakup serangkaian langkah-langkah atau aksi yang harus dijalankan atau diuji untuk memverifikasi fungsi atau fitur yang telah dirancang dalam sistem. Dalam konteks proyek Virtual Produk, skenario-skenario ini mungkin mencakup berbagai situasi penggunaan yang dapat terjadi dalam aplikasi baik segi positif maupun kasus negatif. Sebagai contohnya ketika kita mengakses fitur tersebut namun tiba-tiba internet atau sinyal kita mati, maka akan menampilkan page error yang sebelumnya sudah disepakati.

Dengan membagi skenario menjadi berbagai kategori, tim pengujian dapat mengorganisir dan mengelola pengujian dengan lebih efisien. Hal ini memungkinkan untuk fokus pada pengujian fitur-fitur tertentu dalam setiap kategori, serta memastikan bahwa semua aspek dari fitur Virtual Produk telah diuji secara menyeluruh.

Dengan memiliki skenario yang komprehensif dan terstruktur seperti ini, tim pengembangan dapat memastikan bahwa fitur Virtual Produk telah diuji dengan baik dan siap untuk dirilis kepada pengguna. Hal ini juga membantu dalam menemukan dan memperbaiki *bug* atau masalah potensial sebelum aplikasi diluncurkan secara resmi. Dengan demikian, pembuatan skenario pengujian yang terinci adalah langkah penting dalam memastikan kualitas dan keandalan aplikasi Virtual Produk.

ID	Reporter	Field	Platform	Description	Scenario	QA Step	Status	Priority	Date Status	Date PTC	LHM Test Time	LHM Threshold
							OK	HIGH				
							OK	HIGH				
							OK	HIGH				
							FAIL	MEDIUM				READY TO TEST
							FAIL	HIGH				

**Gambar 3.22 Buglists**

Setelah tahap pengembangan aplikasi selesai, tim *Quality Assurance* lainnya mulai melakukan pengujian pada perangkat Android dan iOS. Pada saat melakukan pengujian apabila ditemukannya *bugs* atau error yang seharusnya tidak terjadi, dan akan mencatat dan memberikan laporan untuk *bug* tersebut kepada divisi yang terkait. Setiap *bug* yang dilaporkan juga memiliki status prioritas. Jika error atau *bug* tersebut telah diinfokan untuk siap dilakukan untuk pengujian lanjutan, akan melakukan *re-test* terhadap *bug* tersebut. Karena proyek ini melibatkan kerjasama dengan tim eksternal, tim QA juga berkomunikasi dengan tim eksternal melalui grup WhatsApp. Grup tersebut digunakan untuk meng-update jika masih terjadi *bug* pada aplikasi. Sebaliknya, tim eksternal juga akan meng-update jika sedang melakukan perbaikan (*fixing*) terhadap *bug* yang dilaporkan.

Komunikasi yang efektif antara tim QA dan tim eksternal melalui grup WhatsApp memastikan bahwa *bug-bug* yang terdeteksi dapat segera diatasi dan proses pengujian dapat berjalan dengan lancar. Hal ini membantu dalam meningkatkan kualitas aplikasi sebelum dirilis secara resmi kepada pengguna akhir.

### **3.3 Kendala yang Ditemukan**

Selama menjalani program kerja magang di PT. Global Loyalty Indonesia, pastinya terdapat beberapa kendala, di antaranya:

- a. Keterbatasan pemahaman terhadap alat-alat yang digunakan oleh perusahaan. Hal ini disebabkan oleh kebaruan dalam penggunaan alat tersebut dan sebelumnya tidak memiliki pengalaman menggunakan alat tersebut, sehingga memerlukan waktu untuk beradaptasi.
- b. Kendala lain yang dihadapi adalah ketika terlibat dalam proyek yang melibatkan kerja sama dengan pihak eksternal. Tantangan utamanya terletak pada komunikasi yang lambat dan kurang responsif dari pihak eksternal. Hal ini menyebabkan pengembangan proyek membutuhkan waktu lebih lama dari yang direncanakan, mengakibatkan penundaan dalam penyelesaian tugas dan mengganggu efisiensi kerja tim.

### **3.4 Solusi atas Kendala yang Ditemukan**

Untuk mengatasi kendala yang dihadapi selama proses program kerja magang di PT. Global Loyalty Indonesia, beberapa solusi yang ditemukan antara lain:

- a. Mendapatkan pengenalan terhadap alat-alat yang digunakan oleh perusahaan dari mentor magang dan dibawah pengawasan supervisor selama satu minggu pertama. Hal ini bertujuan agar dapat melakukan eksplorasi dan menjadi terbiasa dalam menggunakan alat-alat tersebut.

- b. Selain itu, dengan melaksanakan pertemuan rutin secara *online* untuk membahas perkembangan proyek. Dalam pertemuan ini, seluruh pihak terlibat dapat saling berbagi informasi, meninjau kemajuan, dan membahas masalah yang mungkin muncul. Jadwal pertemuan rutin ini akan membantu menjaga arus komunikasi tetap terjaga dan memastikan bahwa semua anggota tim dan pihak eksternal terlibat dalam pengambilan keputusan yang tepat waktu.

