

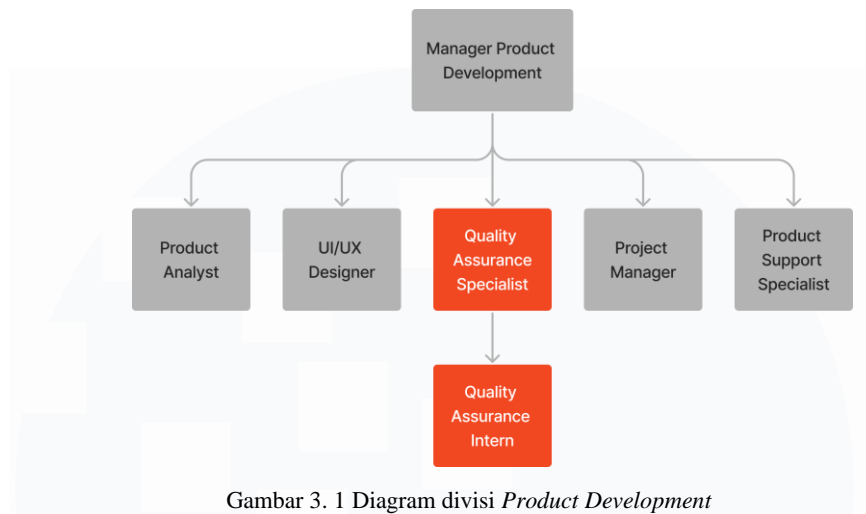
## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Posisi mahasiswa pada saat melakukan program kerja magang yang dilaksanakan di PT Sumber Trijaya Lestari yaitu sebagai *Quality Assurance*. Mahasiswa yang magang dalam posisi *Quality Assurance* dituntut untuk memiliki *skill* dalam melakukan perencanaan dan pembuatan *test* skenario *positif* maupun *negatif*, memahami cara melakukan pencarian data dalam *database*, melakukan *database mapping* (pembuatan *query* untuk pengecekan *database*), dan memahami logika pemrograman. Jobdesc Mahasiswa dalam melakukan *Quality Assurance* adalah sebagai berikut:

- 1) Menentukan dan melakukan pengujian aplikasi yang tepat.
- 2) Mengembangkan dan mendokumentasikan skenario *test case*
- 3) Melakukan pembuatan *query database* dan *mapping database test case*
- 4) Melakukan *reporting* dan *tracking* terhadap *bugs*, *help define*, *implement QA*.
- 5) Berkolaborasi dengan tim *development* dan *product* untuk meningkatkan kualitas produk dan memastikan kebutuhan bisnis.
- 6) Memastikan *produk/service* memenuhi standart kualitas yang ditetapkan termasuk *realibility*, *usability*, dan *performance*.
- 7) Memberi bantuan/arahan pada rekan satu tim dalam menjalankan tanggung jawab utama.
- 8) Melaporkan hasil kerja pada atasan.

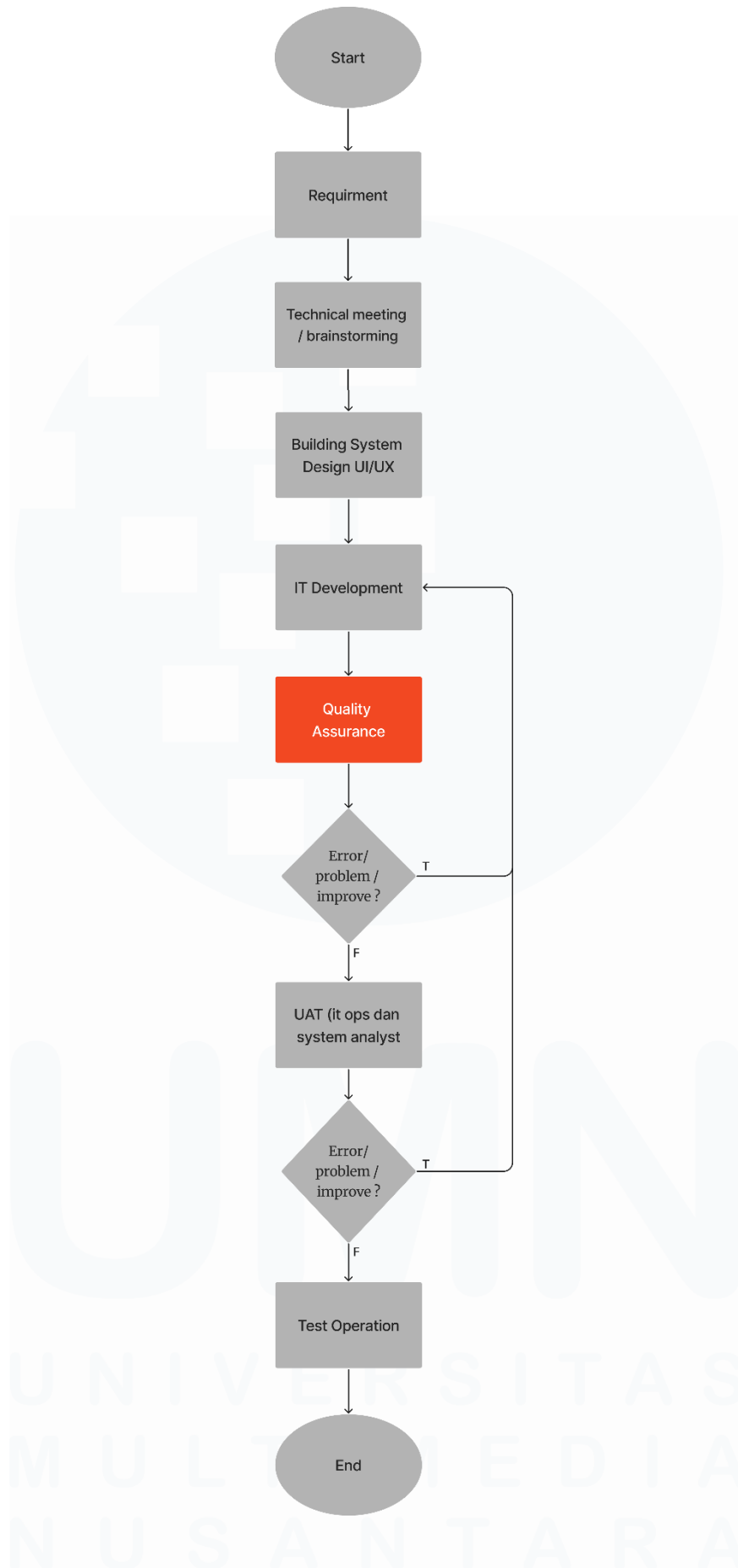


Gambar 3. 1 Diagram divisi *Product Development*

Pada saat melakukan *internship* mahasiswa sebagai *Quality Assurance* berada di bawah divisi *product deveopment*. Divisi *product development* ditunjukkan pada gambar 3.1 adalah divisi yang bertanggung jawab untuk melakukan *development* terhadap *product* mulai dari aplikasi aksemu, aplikasi *management system*, *website dro*, *webiste back office*. Sebagai intern *Quality Assurance* dalam divisi *product development* mahasiswa berkerja dengan banyak subdivisi mulai dari *product analysis* untuk menanyai dan berdiskusi mengenai *system project*, *UI/UX designer* untuk berdiskusi mengenai *design* serta *improvment* untuk *design*, *project manager* untuk pembagian *project* dan rapat mengenai *project*, *Product Support Specialist* untuk berdiskusi mengenai *UAT*. Dalam melakukan pekerjaan mahasiswa melaporkan hasil kerja dia akhir *project* serta *weekly report* ke *manager quality assurace*, *project manager*, dan *Supervisor* dan *product development* yang akan melakukan *review* dan *memberikan masukan* terhadap harsil pekerjaan. Tim *product development* terdiri dari berbagai tim sebagai berikut,

- 1) *Manager tim Product Development*: Pak Bimo
- 2) *Product Analyst*: Mas Adam
- 3) *UI/UX designer*
  - a) *UI/UX Specialist*: Mas Damar
  - b) *UI/UX Intern*: Fardhila Zahra
- 4) *Quality Assurance Specialist*

- a) *Supervisor/QA Specialist*: Mas Himawan
  - b) *Quality Assurance dan Database Specialist* (operational toko): Ko Kelvin Oktavianus
  - c) *Quality Assurance* (Payment and promo Specialties): mbak Putri
  - d) *Quality Assurance (Back Office Specialties harga dan product)*: mbak monika
  - e) *Quality Assurance Intern*: Michael Abhinaya
- 5) *Project manager Officer*: Mbak Novi
- 6) *Product Support Specialist*
- a) Mas Sony
  - b) Mas Fathur
  - c) Mbak Rizky

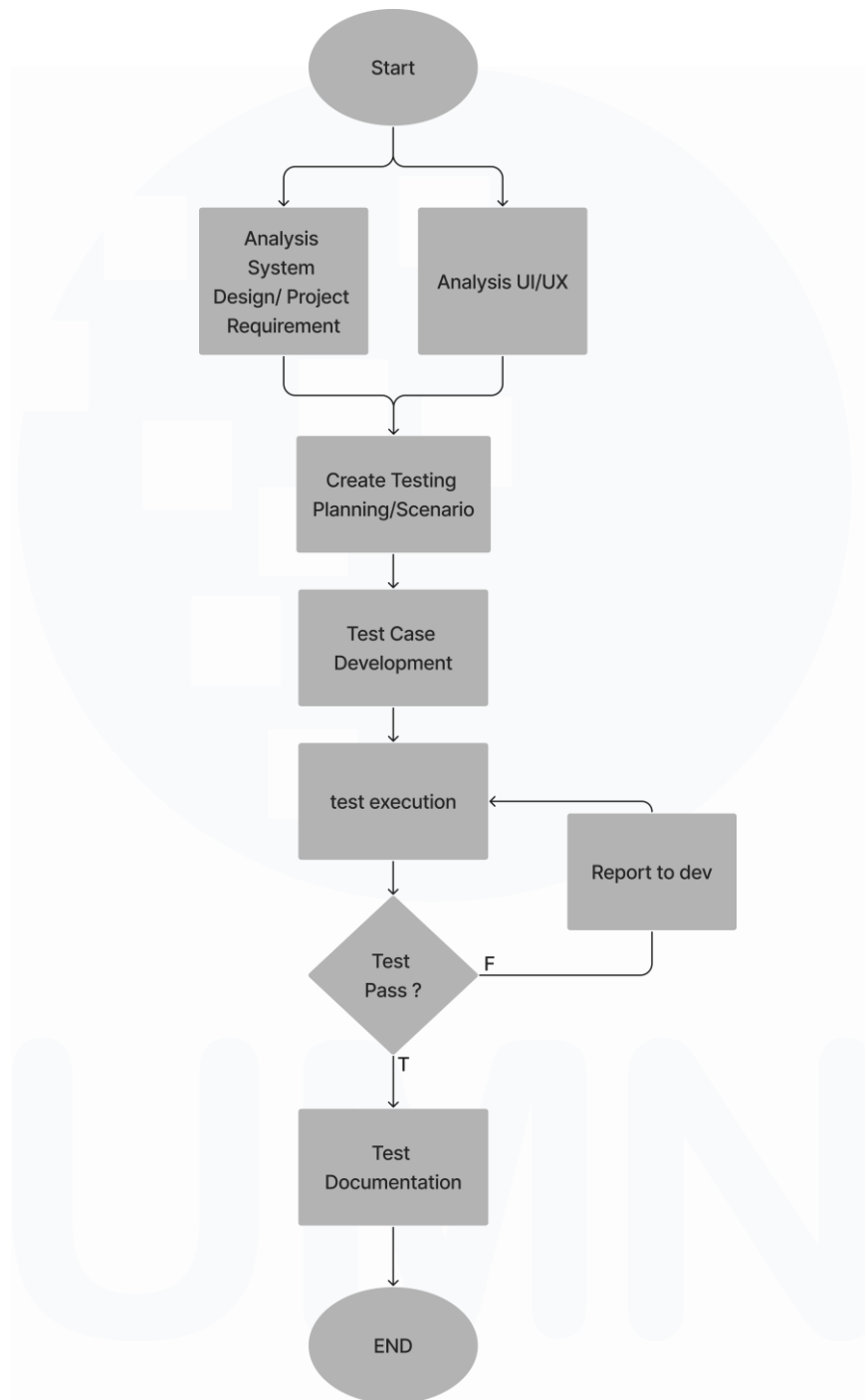


Gambar 3. 2 Flow alur kerja project divisi product development

Alur kerja dari divisi tech *product* dapat digambarkan pada gambar 3.2. *Project* dimulai karena sebuah *requirement* yang dibutuhkan dalam aplikasi atau *website* yang belum ada sebelumnya, permintaan fitur bisa datang dari *general manager*, tim *operation*, tim *finance*, tim *merchandising*. Setelah itu akan dilakukan *Technical Meeting* yang dihadiri oleh *manager UI/UX*, *Project Manager*, *System Analyst*, *IT Operation*, *Quality Assurance*, *Technical meeting* ini akan membahas seluruh syarat, kebutuhan, pelaksanaan dari *project* dan melakukan *brainstorming* bagaimana *project* akan dibuat dan dilaksanakan.

Hasil dari *technical meeting* adalah pembuatan *system design* yang berisi seluruh *background* dari *project*, *business flow* yang *existing* dan yang akan datang, *project scope*, *business rules*, *class diagram*, dan *activity diagram* dari *project*. Selain itu Tim UI/UX akan membuat mockup aplikasi di aplikasi *marvel* dan *figma* berdasarkan hasil berdiskusi dengan *user*. Ketika *system design* dan UI/UX selesai maka akan diberikan ke *IT development* untuk pembuatan aplikasi.





Gambar 3. 3 Flow alur kerja *Quality Assurance*

Ketika *IT Development* sudah selesai melakukan pembuat aplikasi maka tim *Quality Assurance* akan melakukan pengecekan aplikasi secara menyeluruh untuk memastikan aplikasi sesuai dengan kebutuhan *user* dan berkerja secara optimal serta tidak ada kesalahan dari sisi *data*, *UI/UX*, dan

*flow*. Detail proses *Quality Assurance* berkerja tertera pada gambar 3.3. Pertama *Quality Assurance* akan mempelajari *system design* atau *project requirement* dan *mockup* UI/UX. Pembelajaran *system design*, *project requirement* dan ui/ux ini ditujukan agar *Quality Assurance* dapat memahami *background* dari *project* yang diberikan, fitur yang akan ditambahkan, perubahan *business flow*, *minimum viable product*, *activity diagram*, *class diagram*, *wireframe*.

Selanjutnya *Quality Assurance* akan membuat *test planning* atau *testing* skenario berdasarkan analisa dari *system design* atau *project requirement* yang dibuat atau dijelaskan dan *mockup* ui/ux yang sudah selesai dibuat. Pengerjaan *test* skenario ini akan melakukan pengetesan per *section* berdasarkan *system diagram* dan ui sesuai dengan ketentuan pengetesan tertentu. Pembuatan *test* skenario akan dibagi menjadi 2 case yaitu *positive case* dan *negative case*. *Positive case* merupakan sebuah skenario pengetesan dimana aplikasi akan berjalan sesuai skenario yang diharapkan dengan *input* dan data yang positif atau benar, sedangkan *negatif case* merupakan skenario pengetesan untuk data atau input yang tidak valid atau salah dengan tujuan untuk melihat *feedback* dari aplikasi dan *database*. User akan mengisi *test* kolom skenario, hasil yang diharapkan.

Tahap selanjutnya akan dibuat *test case development*, *test case development* merupakan tahapan *user* melakukan *mapping* terhadap data kedalam *database* dan melihat data apa saja yang akan berubah, data apa yang akan dibuuh, data apa saja yang akan masuk dari *project* tersebut ke dalam *database*. Pada tahap ini juga *user* membuat *query* untuk pengecekan, exploring postman atau jenkins apabila ada API dan job yang harus di hit manual. Pada tahap ini dilakukan juga intervensi dan persiapan *database* yang akan dilakukan pengetesan

Selanjutnya akan dilakukan pengetesan atau *test execution* yang dilakukan mengikuti skenario *testing* yang sudah dibuat. *Test* dilakukan per *section* sesuai dengan rencana dan akan dilakukan pengecekan *database* untuk memastikan

input dari aplikasi dengan *database*. Di tahap ini *Quality Assurance* akan mengisi kolom actual result sesuai dengan hasil yang didapat dari pengujian, jika sesuai dengan *expected result* dari *test* skenario dan *database* maka QA pass status untuk skenario tersebut akan dinyatakan OK, namun apabila tidak sesuai akan dinyatakan NOK.

Akan sering dilakukan diskusi dengan *developer* jika terdapat *bug* ataupun improvisasi yang *tester* temukan pada saat melakukan pengetesan. Jika ada *bug* atau kesalahan dalam aplikasi *user* akan memasukan ke dalam list *bug* dan *enhancement* dan melaporkannya ke *developer* yang mengerjakan dan menyiapkan *retest* untuk fitur yang terdampak *bug* mengisi tanggal ditemukan, siapa yang menemukan, deskripsi *bug*, cara mendapatkan *bug*, apa yang harusnya terjadi, dan dokumentasi yang dapat berupa *screenshot* atau video. Jika seluruh pengetesan sudah selesai dilakukan maka *Quality Assurance* akan melakukan *full lifecycle*, yaitu pengetesan secara menyeluruh dari awal sampai akhir untuk memastikan tidak ada fitur yang terdampak atau salah di akhir masa pengetesan.

Dalam pengujian *testing* ini, pengujian dapat dilakukan berpuluh-puluh kali agar mendapatkan hasil yang maksimal. Setiap ada *bug* akan dilakukan *testing* ulang, setiap terdapat perubahan data akan dilakukan *testing* ulang, setiap ada update tentang ui akan dilakukan *testing* ulang. Mahasiswa juga melakukan *testing* ulang akan fitur tertentu atas permintaan dari *developer*. Setelah semua skenario berhasil dilakukan tidak ada *bug* atau *enhancement* dan status nya benar/OK jika tidak benar maka status nya NOK atau *Not OK*, maka mahasiswa melakukan *full lifecycle testing* yang akan melakukan pengujian ulang skenario dari awal sampai akhir, setelah *full lifecycle test* selesai maka *project* dinyatakan QA pass.

Langkah terakhir yang harus dilakukan adalah melakukan pembuatan *Quality Assurance testing* document. QA document berisi tentang segala dokumentasi *testing* yang dilakukan mulai dari *background*, *business rule*, id yang dipakai untuk melakukan *testing*, *database* yang dipakai dan tabel yang



dipakai beserta column, serta *flow case* yang dibuat selama melakukan *testing* akan di dokumentasi sebagai refrensi untuk uat dan *project* mendatang.

Selanjutnya akan dilaksanakan *User Acceptance Testing* (UAT) yang aplikasi memasuki tahap beta. Pada tahap ini akan dilakukan pengetesan oleh tim *Product Support Specialist* untuk pengetesan terakhir aplikasi berdasarkan skenario dan document yang sudah dibuat sebelumnya. *Test UAT* ini juga berlaku ke vendor apabila PT Sumber Trijaya Lestari menggunakan vendor maka tes UAT dari vendor juga akan dikerjakan di tahap ini. Jika ada permasalahan atau improvement saat melakukan UAT maka *project* akan kembali ke tahap *development* dan akan dilakukan *test* ulang untuk *project* tersebut. Namun apabila UAT sukses dilaksanakan maka *project* akan berubah status menjadi live atau TO yang membuat data aplikasi akan berubah menjadi data *production* atau data nyata yang digunakan oleh *user* dan *project* dinyatakan selesai. Akhri dari kegiatan *testing* akan dilakukan *weekly report* setiap awal minggu ke *manager* untuk mengetahui progress *Quality Assurance testing*. Seluruh hasil pekerjaan *testing* dan *documentation* akan diperiksa oleh *manager Quality Assurance*

### **3.2 Tugas dan Uraian Kerja Magang**

Kerja magang pada PT Sumber Trijaya Lestari dimulai sejak tanggal 16 Febuari 2023 hingga 20 Juni 2023. Terdapat beberapa *project* dan *side project* yang dikerjakan oeh mahasiswa yang meliputi kegiatan *onboarding*, *project* login dan rekap absensi AMD, *project* pembayaran QRIS, *project* kunjungan ac/am, *project* setoran, serta *side project* UAT vendor pembayaran, dan *test* QRIS untuk Bank Indonesia. Tabel 3.1 merupakan gambaran besar dari kegiatan yang dilaksanakan selama periode magang di PT Sumber Trijaya Lerstari.

Seluruh pengerjaaan *project* di perusahaan PT Sumber Trijaya Lestari mengikuti Standard Operating Procedure yang sudah ditepakan perusahaan. Seluruh pengetesan dalam *project* yang dilakukan menggunakan metode

manual regression *testing* dengan *system* pengerjaan SOP yang digambarkan pada gambar 3.3.

Tabel 3. 1 Waktu Pelaksanaan Magang Perusahaan

Nomor	Judul Aktivitas	Mulai	Selesai
<b>1</b>	<b>Onboarding</b>		
1.1	Pengenalan tim, <i>software</i> , sistem perusahaan, dan alur kerja	16 Febuari 2023	20 Febuari 2023
1.2	Pengenalan terhadap <i>database</i> dan <i>query</i> perusahaan	16 Febuari 2023	17 Febuari 2023
1.2	Pemahaman proses bisnis dan <i>database</i> perusahaan	20 Febuari 2023	20 Febuari 2023
<b>2</b>	<b>Project Login dan Rekap Absensi AMD (<i>website</i>)</b>		
2.1	Identifikasi Requirement <i>project</i> rekap absensi dan login AMD	21 Febuari 2023	21 Febuari 2023
2.2	Pembuatan skenario <i>testing</i> , <i>mapping database</i>	21 Febuari 2023	23 Febuari 2023
2.3	<i>Testing</i> aplikasi	22 Febuari 2023	02 Maret 2023
2.4	QA <i>Documentation</i>	02 Maret 2023	03 Maret 2023
2.5	Improvement from UAT	07 Maret 2023	08 Maret 2023
<b>3</b>	<b>Project Pembayaran Baru QRIS</b>		
3.1	Identifikasi Requirement <i>project</i> pembayaran qris	10 Maret 2023	14 Maret 2023
3.2	pembuatan skenario <i>testing front office</i> , melakukan <i>mapping database</i>	13 Maret 2023	17 Maret 2023
3.3	pembuatan skenario <i>testing back office</i> . Melakukan <i>mapping database</i>	13 Maret 2023	17 Maret 2023

<b>Nomor</b>	<b>Judul Aktivitas</b>	<b>Mulai</b>	<b>Selesai</b>
3.4	<i>testing</i> aplikasi QRIS dari sisi <i>front office</i>	20 Maret 2023	03 April 2023
3.5	<i>testing</i> promo dari sisi <i>front office</i> maupun <i>back office</i>	20 Maret 2023	03 April 2023
3.6	<i>testing</i> aplikasi QRIS dari sisi <i>back office</i>	20 Maret 2023	03 April 2023
3.8	Enhancement fitur	03 April 2023	04 April 2023
3.7	Full <i>lifecycle testing</i>	04 April 2023	05 April 2023
3.9	Membuat QA <i>Documentation project</i> QRIS	06 April 2023	10 April 2023
3.10	Melakukan UAT dengan vendor pengiriman	05 April 2023	05 April 2023
3.11	Melakukan UAT dengan <i>Operational Manager</i>	12 Mei 2023	12 Mei 2023
3.12	Melakukan <i>Test Internal</i> ke Toko	24 Mei 2023	24 Mei 2023
<b>4</b>	<b><i>Side Project UAT Vendor Pembayaran</i></b>		
4.1	Pengerjaan UAT Vendor pembayaran	12 April 2023	17 April 2023
<b>5</b>	<b><i>Project Kunjungan AC/AM</i></b>		
5.1	Identifikasi <i>Project Requirement</i> Kunjungan AC./AM	20 April 2023	21 April 2023
5.2	Pembuatan <i>Test Scenario</i> dan <i>database mapping</i>	24 April 2023	24 Mei 2023
5.3	Pengetesan awal aplikasi	15 Mei 2023	16 Mei 2023
<b>6</b>	<b><i>Side project tes fungsionalitas SNAP Bank Indonesia</i></b>		

Nomor	Judul Aktivitas	Mulai	Selesai
6.1	Training Coaching Clinic SNAP	02 Mei 2023	02 Mei 2023
6.2	Pengerjaan <i>Test</i> Fungsionalitas	02 Mei 2023 16 Mei 2023	03 Mei 2023 16 Mei 2023
<b>7</b>	<b><i>Project Enhancement Setoran</i></b>		
7.1	Rapat dengan tim untuk identifikasi <i>project</i> Enhancement Setoran	5 Juni 2023	5 Juni 2023
7.2	Pembuatan <i>test</i> Skenario	5 Juni 2023	9 Juni 2023
7.3	<i>Database mapping</i> dan pembuatan <i>query</i>	6 Juni 2023	9 juni 2023

### 3.2.1 Onboarding

#### 3.2.1.1 Pengenalan tim, *software*, dan alur kerja

Pada awal dari dilakukannya magang diadakan meeting dengan seluruh Tim *Quality Assurance* untuk melakukan onboarding dan familiriasasi dari keseluruhan *system* dan cara kerja perusahaan. Ditahap ini dijelaskan berbagai sistem yang ada di PT Sumber Trijaya Lestari, sistem *front office* untuk aplikasi aksesmu ke *customer*, *back office* untuk manajemen pegawai aksesmu, Aksesmu memiliki penentuan harga, stok, promo.

PT Sumber Trijaya Lestari memiliki 3 tahap dalam mengembangkan aplikasi yaitu staging, beta, dan *production*. Tim *Quality Assurance* akan melakukan pengetesan fitur dan aplikasi dalam tahapan staging perusahaan, yang memiliki server dan *database* yang terpisah dari *database production*.

*Software* yang digunakan pada Tim *Quality Assurance* untuk melakukan pengerjaan *project* aksesmu ada berbagai macam yaitu:

1) Marvel



Gambar 3. 4 Logo marvel

Sumber: (<https://marvelapp.com/>)

Marvel ditunjukkan pada gambar 3.4 [9] adalah satu dari dua *software* yang digunakan oleh Tim UI/UX untuk pembuatan Mockup Figma. *Quality Assurance* dapat melihat langsung tampilan mockup atau *prototype* dari *project* yang dibuat dari link yang diberikan oleh tim UI/UX untuk dianalisa dan melihat flow dari UI/UX.

2) Figma



Gambar 3. 5 Logo figma

Sumber: (<https://www.figma.com/>)

Figma ditunjukkan pada gambar 3.5 [10] adalah *software* pembuat UI/UX yang digunakan juga oleh tim UI/UX untuk melakukan pembuatan mockup aplikasi. Mockup akan diberikan dalam bentuk *link* dan akan dapat dilihat lebih mudah *prototype* dan *flow* dari *link* aplikasi

### 3) Jenkins



Gambar 3. 6 Logo jenkins

Sumber: (<https://www.jenkins.io/>)

Jenkins ditunjukkan pada gambar 3.6 [11] adalah *software* pembantu *developer* yang digunakan untuk melakukan integrasi otomatisasi sebuah fungsi atau *deployment* dengan bahasa java dan bersifat *open-source*. Jenkins memudahkan *developer* untuk membuat sebuah otomatisasi sebuah fungsi. Jika *Quality Assurance* membutuhkan menjalankan suatu fungsi untuk pengetesan, dapat dilakukan *restart service* atau *running job* ke fungsi jenkins yang dibutuhkan tanpa harus meminta bantuan *developer*.

### 4) Postman



Gambar 3. 7 Logo postman

Sumber: (<https://www.postman.com/>)

Postman ditunjukkan pada gambar 3.7 [12] adalah *software* khusus api yang digunakan untuk melakukan pembuatan, pengiriman, dan permintaan berbagai *Application Programming Interface API* dari *developer*.

Postman membantu *developer* dan *Quality Assurance* untuk memudahkan tahap pembuatan API, menyediakan berbagai tool, dan memungkinkan kolaborasi dan penyimpanan API untuk kebutuhan operation API perusahaan. Postman digunakan oleh *Quality Assurance* untuk melakukan hit api untuk melakukan perubahan fungsi atau data didalam pengecekan aplikasi jika dibutuhkan.

Pada tahap ini juga dilakukan percobaan aplikasi dasar yang paling sering dipakai didalam perusahaan yaitu *front office* aksesmu dan *back office* aksesmu management system. Dilakukan berbagai percobaan menggunakan staging mulai dari pembuatan akun *member*, melakukan pesanan dengan berbagai opsi pembayaran, melakukan percobaan berbagai promo, melakukan explore lebih dalam terhadap fitur yang dapat dilakukan didalam aplikasi untuk memperdalam pengetahuan dan *bussiness knowledge*.

#### 3.2.1.2 Pengenalan terhadap *database* dan *query* perusahaan

Pada tahap ini dilakukan pengenalan lebih dalam terhadap berbagai macam *database* dasar yang terdapat pada PT Sumber Trijaya Lestari. Mahasiswa mempelajari *database* dan melihat data yang ada selagi menerapkan *flow business* perusahaan mulai dari pembuatan akun *customer* sampai dengan pengiriman barang selesai di sisi *officer*.

Terdapat 2 *software database* yang digunakan oleh perusahaan untuk melakukan penambahan, perubahan, dan pengecekan data:

### 1) Robo3T Mongo

Robo3T atau yang dikenal juga dengan RoboMongo adalah *software* manajemen *database* mongo yang pada penggunaan *shell*. Robo3t digunakan untuk melakukan akses *database* Mongo yang merupakan *database* yang tidak menggunakan struktur *relational database* (NoSQL). *Database* Mongo diperuntukan untuk seluruh data *member* dan *officer*.

### 2) DBeaver postgreSQL

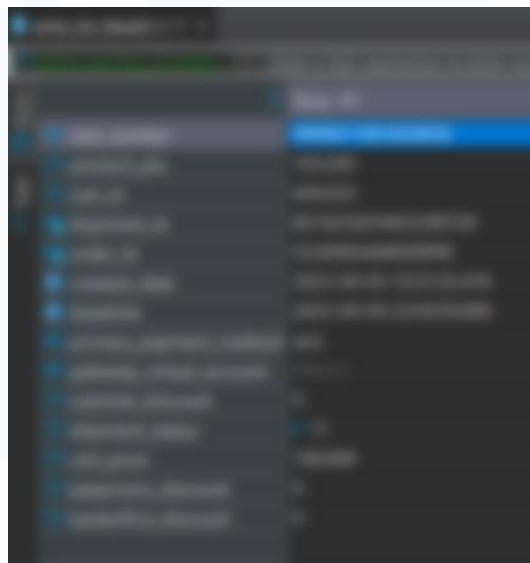
Dbeaver adalah aplikasi *sql software* dan administrasi *database*. Aplikasi ini menggunakan *interface* pemrograman JDBC untuk berinteraksi dengan *database* menggunakan driver yang disediakan. Dbeaver bersifat *cross platfrom* yang *compatible* dengan berbagai tipe *database*. *Database* yang digunakan didalam Dbeaver untuk PT Sumber Trijaya lestari adalah PostgreSQL. PostgreSQL adalah *Relational Database Management System* (RDBMS) yang digunakan oleh PT Sumber Trijaya Lestari untuk data utama perusahaan mulai dari transaksi, *promo*, stok, dan *log*.



Gambar 3. 8 Pembuatan *query* training *database*



Pada tahap ini dilakukan familiarisasi *database* dengan melakukan pengecekan data selagi mencoba aplikasi *front office* dan *back office*. Selain itu, dilakukan juga pelatihan pembuatan *query* seperti gambar 3.8 yang memiliki tujuan agar *quality assurance* memiliki pemahaman yang lebih dalam dan familiriasi terhadap *enviroment* database yang digunakan didalam perusahaan.



Gambar 3. 9 Hasil *query* training

Dari *query* yang dibuat dapat dilihat perubahan dan pengecekan data yang masuk untuk aplikasi *front office* dan *back office*. Gambar 3.9 merupakan gambaran hasil *query* yang dibuat untuk mengecek data aplikasi.

### 3.2.1.3 Pemahaman proses bisnis dan *database* perusahaan

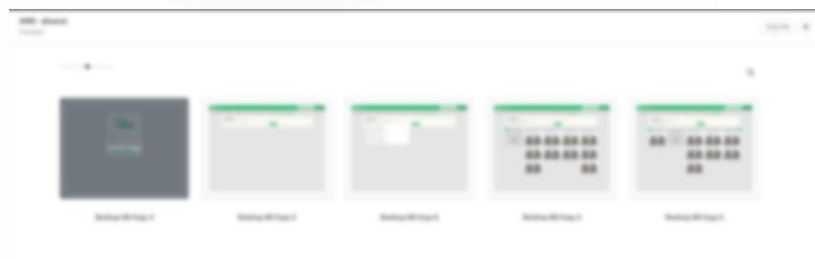
Pada tahap ini manager *product development* menjelaskan kembali secara detail terkait dengan proses bisnis PT Sumber Trijaya Lestari. Hal ini ditujukan agar mahasiswa dapat lebih memahami keseluruhan proses bisnis yang dilakukan oleh perusahaan. Materi yang dipaparkan mulai dari cara aplikasi

berkerja, SOP perusahaan, serta bagaimana koneksi setiap *database* dan data apa saja yang tersimpan.

### 3.2.2 Project Login dan Rekap Absensi AMD (*website*)

#### 3.2.2.1 Melakukan Identifikasi Requirement *project* rekap absensi dan login AMD

*Project* Login dan Rekap Absensi AMD merupakan *project* untuk melakukan pengenalan *database* dan *system* cara kerja perusahaan secara lebih mendalam. *Project* ini memiliki tujuan agar *operational manager*, *area coordinator*, *area manager* dapat melakukan rekap absensi bawahan melalui *website Aksesmu Management Dashboard* (AMD). Website AMD dapat diakses dari manapun, namun akses tetap dibatasi berdasarkan role karyawan. *Project* ini adalah *project* lanjutan dari fitur rekap absensi yang sudah ada dalam aplikasi *back office* Aksesmu management system (AMS).



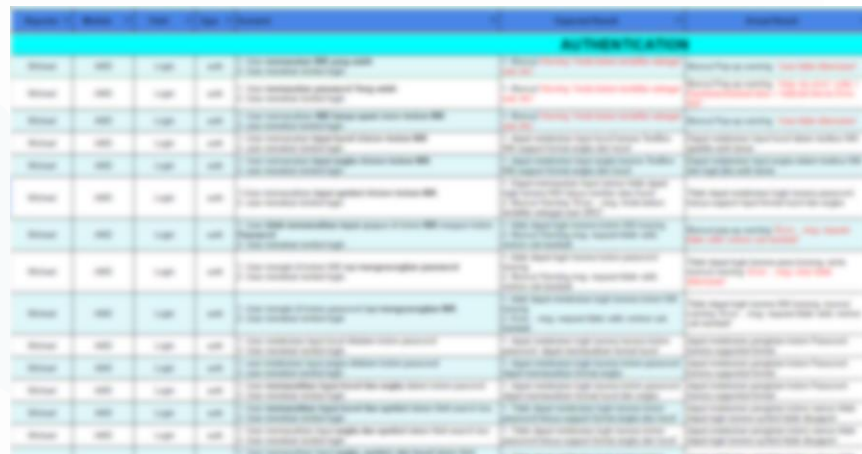
Gambar 3. 10 UI/UX Rekap Absensi

*Quality Assurance* akan mempelajari *mockup website* yang telah dibuat dan diupload pada aplikasi marvel seperti Gambar 3.10 untuk mempelajari flow dan melihat potensi skenario yang dapat dibuat untuk kebutuhan pengetesan aplikasi. Selain itu *Quality Assurance* juga mempelajari pembatasan fitur atau business rules dari fitur yang dapat diakses oleh *officer* perusahaan berdasarkan diskusi dengan *IT developer* dan

*manager* dikarenakan tidak ada *system design* khusus untuk *project*.

### 3.2.2.2 Melakukan pembuatan skenario *testing*, Melakukan *mapping Database*

Setelah mempelajari mockup dan *database* dari perusahaan dari hasil diskusi, dibuat *test* skenario berdasarkan mockup, *business rule*, serta hasil diskusi dengan *developer*. *Test* skenario yang dibuat dibagi menjadi tiga bagian pengetesan. *Test* skenario dibuat untuk melakukan pengecekan tiga bagian utama yaitu UI/UX (mengecek icon, tampilan ui, logo, box pengisian, input pengisian, dan *user experiece* dari pengguna), *database* (melakukan pengecekan data mulai dari input, display, intervensi data, penyiapan data untuk kebutuhan pengetesan), *flow* dan *limit* aplikasi (melakukan pengecekan apa yang dapat dan tidak dapat *user* lakukan, untuk menemukan celah *bug* dan menentukan apakah limit dari *business rule* sudah sesuai dengan apa yang dijelaskan oleh *developer*).



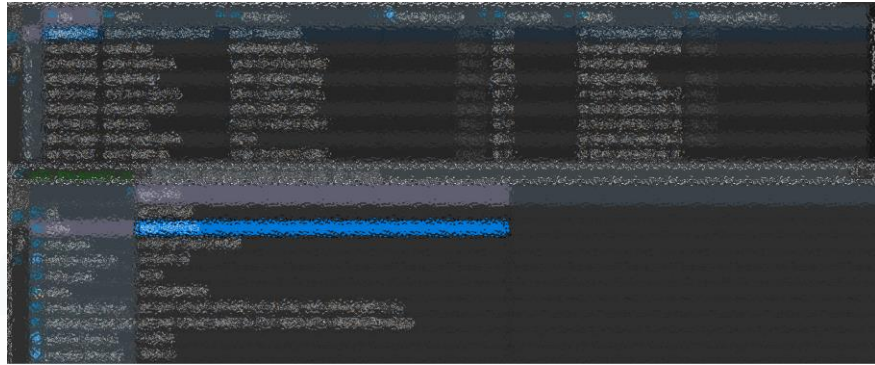
The image shows a screenshot of a test scenario table. The table has several columns, including 'No', 'Uraian', 'Langkah', 'Data', 'Kondisi', and 'Hasil'. The content is mostly illegible due to blurring, but it appears to be a detailed test plan for a login process, listing various user roles and their expected actions and outcomes.

Gambar 3. 11 *Test* skenario bagian satu

Hal yang pertama yang dilakukan adalah melakukan pembuatan skenario bagian satu, yang berisi *login* dari role yang dapat melakukan akses terhadap *webpage* AMD yang ditunjukkan pada gambar 3.11. Secara *general* hampir semua *role* dalam kategori *managerial* dapat







Gambar 3. 15 Hasil pembuatan *query* untuk rekap absensi dan login

Gambar 3.15 merupakan gambaran hasil pembuatan *query* yang dibuat untuk menghasilkan data-data mengenai rekap absensi karyawan dengan kondisi skenario pengetesan.

### 3.2.2.3 Melakukan *testing* aplikasi

Proses *testing* dilakukan bersamaan dengan pembuatan *test* skenario dan *mapping database* dikarenakan pada saat *website* dibuat, *testing* sudah dapat dilakukan. *Testing* aplikasi dilakukan sesuai dengan skenario yang telah dibuat dan dilakukan juga pengecekan dari ui/ux, *database*, dan flow dari skenario.

A screenshot of a testing sheet for absence records. It is a table with multiple columns and rows. The columns include test case ID, test case description, test data, test steps, test results, and status. The status column shows 'OK' for several rows, indicating successful test results. The table is highlighted in light blue.

Gambar 3. 16 Contoh *sheet* pengujian rekap absensi

Proses *testing* dimulai berdasarkan pembuatan skenario bagian awal hingga bagian akhir. Jika pengujian sudah sesuai, maka QA pass status akan berubah menjadi OK seperti gambar 3.16.

ID	Date	Description	Steps to Reproduce	Expected Results	Actual Results	Status
1	2020-03-01	...	...	...	...	Open
2	2020-03-01	...	...	...	...	Open
3	2020-03-01	...	...	...	...	Open
4	2020-03-01	...	...	...	...	Open
5	2020-03-01	...	...	...	...	Open
6	2020-03-01	...	...	...	...	Open
7	2020-03-01	...	...	...	...	Open
8	2020-03-01	...	...	...	...	Open
9	2020-03-01	...	...	...	...	Open
10	2020-03-01	...	...	...	...	Open
11	2020-03-01	...	...	...	...	Open
12	2020-03-01	...	...	...	...	Open
13	2020-03-01	...	...	...	...	Open
14	2020-03-01	...	...	...	...	Open
15	2020-03-01	...	...	...	...	Open
16	2020-03-01	...	...	...	...	Open
17	2020-03-01	...	...	...	...	Open
18	2020-03-01	...	...	...	...	Open
19	2020-03-01	...	...	...	...	Open
20	2020-03-01	...	...	...	...	Open

Gambar 3. 17 Sheet tampilan *bug reporting* Rekap Absensi

Pada saat melakukan *testing* ini mahasiswa menemukan berbagai macam *bug* mulai dari ui/ux belum dapat login ke dalam *website*, *placement* dan margin dari tampilan, *bug* dari data yang ditampilkan tercampur, maupun tidak sesuai urutan. Setiap *bug* yang ditemukan akan dilaporkan pada sheet *bug/enhancement* disini *test Quality Assurance* menemukan berbagai macam *bug* mengenai ui error sampai dengan data yang error. Semua *bug* akan didokumentasikan dan disimpan dengan *screenshot* atau video ke dalam google drive seperti gambar 3.17.

Setelah melakukan *Bug fixing* dan semua aplikasi berjalan dengan skenario akan dilakukan *full lifecycle testing* terakhir. Hal ini untuk memastikan tidak ada fitur lain yang berubah atau rusak karena *bug fixing* yang dilakukan oleh *developer*.

### 3.2.2.4 Membuat QA Documentation



Gambar 3. 18 Contoh dokumentasi *project* rekap absensi

Setelah menyatakan QA pass, akan dibuat *QA documentation* yang berisi mengenai seluruh dokumentasi dari *testing* yang dilakukan untuk *project* tersebut. Tujuan dari pembuatan *documentation* ini adalah ketika



ada perubahan *project* atau *improvement project* pada saat yang akan datang, terdapat laporan mengenai *project* sebagai referensi. Gambar 3.18 menunjukkan isi dari QA *documentation* meliputi *background* dari *project*, *business rule* dari *project*, parameter yang diperlukan, proses dari rekap absensi, data yang digunakan untuk melakukan *testing*, *flow positive* dari *testing*, *flow negative* dari *testing*, *coding query* untuk pengecekan data flow.

### 3.2.2.5 Improvement from UAT

Proses terakhir yang dilakukan dalam *project* ini adalah *user acceptance testing* untuk melakukan pengetesan fitur baru yang disematkan dalam *project* dengan data *production* namun masih dalam fase beta. Proses UAT dari *project* rekap absensi AMD oleh tim IT *support* dan *operational manager* untuk melakukan pengetesan internal fitur dari *project*, hasilnya terdapat penambahan fitur dan restriksi sesuai dengan keinginan pengguna aplikasi.

Test Case ID	Test Case Description	Test Case Data	Test Case Expected Results	Test Case Actual Results	Test Case Status
TC001	Test Case Description	Test Case Data	Test Case Expected Results	Test Case Actual Results	Pass
TC002	Test Case Description	Test Case Data	Test Case Expected Results	Test Case Actual Results	Pass
TC003	Test Case Description	Test Case Data	Test Case Expected Results	Test Case Actual Results	Pass
TC004	Test Case Description	Test Case Data	Test Case Expected Results	Test Case Actual Results	Pass

Gambar 3. 19 Tambahan *Test Scenario Enhancement*

Dikarenakan terdapat penambahan fitur pada saat uat maka Quality Assurance akan membuat *test* skenario tambahan ditunjukkan pada gambar 3.19 untuk melakukan pengetesan terhadap fitur dan restriksi baru yang akan diimplementasikan.

Business Rule ID	Business Rule Description	Business Rule Data	Business Rule Expected Results	Business Rule Actual Results	Business Rule Status
BR001	Business Rule Description	Business Rule Data	Business Rule Expected Results	Business Rule Actual Results	Pass
BR002	Business Rule Description	Business Rule Data	Business Rule Expected Results	Business Rule Actual Results	Pass

Gambar 3. 20 Tambahan *business rule* Untuk melihat absensi



Setelah pengetesan kembali untuk fitur berhasil, akan dilakukan pencacatan tambahan dari fitur dan restriksi fitur di dalam dokumen rekap presensi AMD yang dapat dilihat pada gambar 3.20.

### 3.2.3 *Project* pembayaran baru QRIS

#### 3.2.3.1 Melakukan Identifikasi Requirement *project* pembayaran qris

*Project* selanjutnya yang dikerjakan adalah *project* pembayaran baru *customer* Aksesmu yaitu integrasi *Quick Response Code Indonesian* (QRIS).

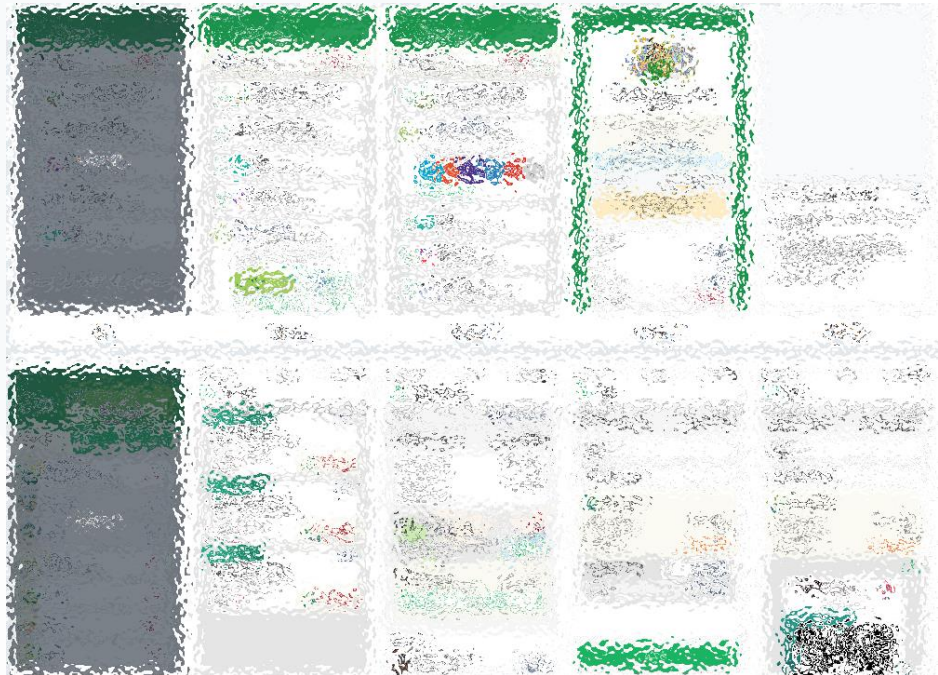


Gambar 3. 21 *Business requirement* QRIS

*Project* pembayaran baru diintegrasikan untuk menambah metode pembayaran baru non tunai dalam *system* aksesmu dan meningkatkan cakupan kapabilitas dan keamanan pembayaran untuk segmentasi *customer* yang lebih luas. Pada tahap ini dilakukan analisa *system design project* pembayaran baru yang dikirimkan oleh *project manager* ataupun supervisor. gambar 3.21 menunjukkan *system design* dari *project* pembayaran QRIS

Dari *system design* dapat dipelajari *class diagram*, *activity diagram*, dan *wireframe*, *business requirement* dan *Minimum Viable Product* dari aplikasi untuk mengetahui bagaimana flow keseluruhan dari aplikasi, batasan apa saja yang harus diterapkan, apa saja yang akan ditampilkan dalam aplikasi, dan *database* serta column apa saja

yang akan dipakai dalam aplikasi. Selain itu dijelaskan juga untuk vendor yang dipakai untuk melakukan pembayaran dari aksesmu ke Bank Indonesia menggunakan vendor pembayaran dimana seluruh data mengenai pembayaran QRIS akan datang dari vendor pembayaran ke *database* aksesmu.



Gambar 3. 22 UI/UX aplikasi front office dan back office QRIS

Pada tahap ini juga dipelajari UI/UX dari link marvel yang diberikan oleh tim UI/UX untuk desain tampilan pembayaran QRIS dari sisi Aksesmu (AMA) dan Aksesmu Management System (AMS) ditunjukkan pada gambar 3.22. Dari UI/UX yang diberikan dapat dipelajari apa saja tampilan yang berubah dalam aplikasi AMA dan AMS serta dapat melihat data yang akan ditampilkan dalam aplikasi nantinya. Mockup UI/UX memudahkan *Quality Assurance* melakukan pembuatan *test* skenario *front office* dan *back office* serta data yang harus ditambahkan dalam *test* skenario tersebut.

Pada fase ini juga diberikan dijelaskan oleh *developer database* yang akan dipakai untuk *project* ini. Terdapat 4 *database* utama yang dipakai untuk *project* ini dan beberapa penambahan column untuk

mengakomodasi pembayaran QRIS yang mengambil data QR dari Vendor pembayaran.

### 3.2.3.2 Melakukan pembuatan skenario *testing front office*, melakukan *mapping database*



Gambar 3. 23 Sheet skenario *testing front office* QRIS

Pertama dilakukan pembuatan *test* skenario dari sisi *front office* yaitu aplikasi aksesmu yang dibagi menjadi dua bagian *basic member* dan *full member*. Gambar 3.23 adalah *Scenario test* yang dibuat menggunakan spreadsheet untuk sisi *front office* dalam melakukan pengecekan tambahan UI aplikasi untuk menampilkan pembayaran QRIS pada saat melakukan pemilihan pembayaran. Selain itu, dibuat juga pembuatan *test* skenario untuk mengakomodasi pengetesan *thankyou page* dimana akan menampilkan *thankyou page* dari qris sendiri beserta total yang harus dibayar.

Setelah *test* skenario selesai dibuat, *quality assurance* melakukan *mapping database* untuk mengetahui data apa yang terdapat dalam aplikasi dan perubahan data yang terjadi. Setiap skenario yang dibuat yang melibatkan data akan dicatat dalam *test* skenario yang meliputi *database* yang digunakan, table yang dipakai, dan *column* data yang berubah atau terpakai.



mengecek database terkait system business dan QRIS, indikasi pengiriman data ke vendor.

### 3.2.3.4 Melakukan *testing* aplikasi QRIS dari sisi *front office*



Gambar 3. 26 Tampilan spreadsheet *testing front office* QRIS

Untuk pengujian aplikasi QRIS dilakukan dengan mengikuti *testing* skenario yang dibuat selagi melakukan perbaikan dan penambahan skenario selagi melakukan pengujian. Gambar 3.26 adalah tampilan pengujian *front office* yang dilakukan untuk mengecek apakah tampilan yang ditampilkan di sisi *customer* sudah sesuai dengan *mockup*, data serta *system*. Selain itu dilakukan juga pengecekan dari sisi *database* dengan melakukan run *query* yang sudah dibuat pada saat *mapping database*. Pengecekan juga akan meliputi pengecekan log *payload* aplikasi untuk melihat data yang dikirimkan dari atau ke aplikasi.



Gambar 3. 27 Contoh pelaporan Bug sisi *front office* QRIS

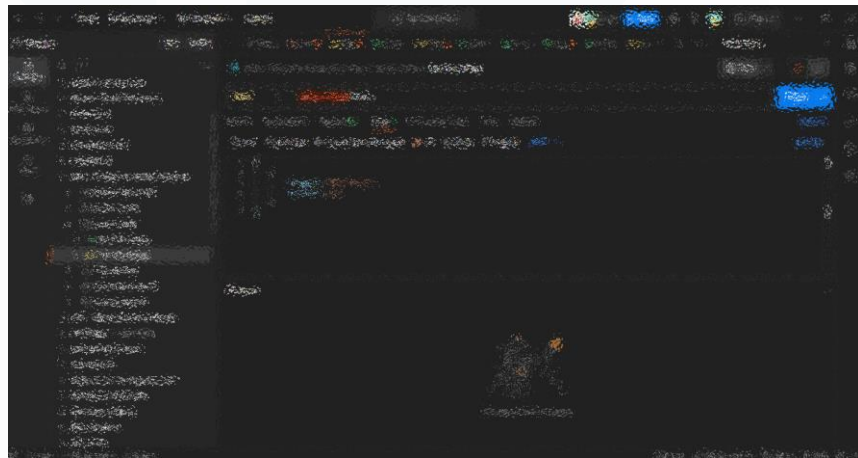
Jika ditemukan *bug* didalam aplikasi akan dilakukan pencacatan berdasarkan log *test* aplikasi dan dokumentasi didalam spreadsheet *buglist* seperti gambar 3.27 dan *memberitahu developer* mengenai



*bug* yang ada untuk segera diperbaiki lalu akan dilakukan *testing* ulang untuk menentukan apakah fixing berhasil.

### 3.2.3.5 Melakukan *testing* aplikasi QRIS dari sisi *back office*

Selanjutnya dilakukan berbagai pengetesan dari sisi *Back Office* sesuai tahapan yang sudah diselesaikan oleh *developer*. Seluruh pengetesan akan dilakukan sesuai dengan *test* skenario yang sudah dibuat selagi melihat kemungkinan penambahan *test* skenario. Selain itu dilakukan juga pengecekan *database* untuk melihat apakah data sudah sesuai dengan melakukan run *query* yang sudah dibuat sebelumnya. Pengecekan juga akan meliputi pengecekan log *payload* aplikasi untuk melihat data yang dikirimkan dari atau ke aplikasi.



Gambar 3. 28 Contoh postman yang digunakan untuk pengetesan QRIS

Gambar 3.28 adalah salah satu pengetesan yang dilakukan dari sisi *back office* qris untuk melihat deadline pembayaran. Postman dilakukan untuk melakukan pengetesan pembayaran dengan

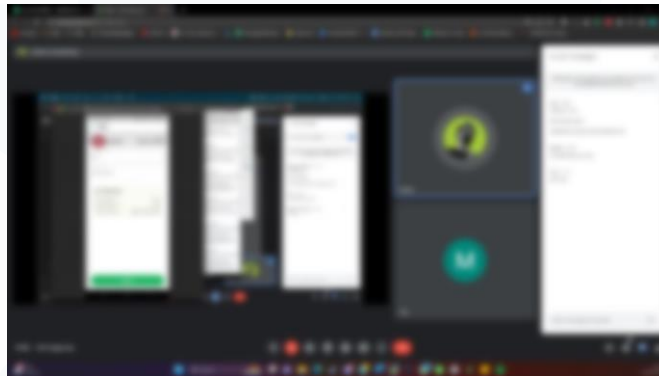
memasukan nomor data ke dalam postman dan di run ke server staging.



ID	Category	Priority	Description	Reporter	Status	Assignee
000001	Payment	High	Payment QRIS not working	John Doe	Open	John Doe
000002	Payment	Medium	QRIS code not visible	Jane Smith	Open	Jane Smith
000003	Payment	Low	QRIS code not working	John Doe	Open	John Doe
000004	Payment	High	Payment QRIS not working	Jane Smith	Open	Jane Smith
000005	Payment	Medium	QRIS code not visible	John Doe	Open	John Doe
000006	Payment	Low	QRIS code not working	Jane Smith	Open	Jane Smith
000007	Payment	High	Payment QRIS not working	John Doe	Open	John Doe
000008	Payment	Medium	QRIS code not visible	Jane Smith	Open	Jane Smith
000009	Payment	Low	QRIS code not working	John Doe	Open	John Doe
000010	Payment	High	Payment QRIS not working	Jane Smith	Open	Jane Smith

Gambar 3. 29 Sheet *buglist* untuk *project* QRIS

Saat melakukan pengetesan ditemukan berbagai *bug* yang muncul di aplikasi yang dicatat dalam *buglist* dan dilakukan dokumentasi cara menemukan *bug* agar dapat diperbaiki oleh *developer* yang ditunjukkan pada gambar 3.29.



Gambar 3. 30 Rapat dengan dev untuk *fixing bug* QRIS

Gambar 3.30 adalah gambaran dalam proses perbaikan dan *reporting bug* ke *developer*, dilakukan pertemuan menggunakan google meet untuk melakukan pembahasan *bug* atau error yang terjadi dan diskusikan *expected* skenario yang terjadi.

#### 3.2.3.6 Melakukan *testing* promo dari sisi *front office* maupun *back office*

Salah satu hal yang harus dilakukan dari *project* ini adalah melakukan pengetesan pembayaran qris terhadap setiap promo yang ada dalam aplikasi aksesmu. Resiko paling besar dalam penambahan fitur pembayaran adalah bentroknnya *project* dengan fitur lain yang krusial seperti promo. Oleh karena itu, setelah *testing* aplikasi dari

sisi *front office* dan *back office* sesuai dengan ekspektasi, dilakukan *testing* promo secara mendalam dengan pembayaran qris. Pengetesan promo ini dilakukan dengan melakukan pengecekan *query* dan melakukan pembuatan promo. Lalu akan dibuat serangkaian pengetesan dengan menggunakan berbagai macam promo yang ada di dalam *system* dan akan di cek menggunakan *query*.



Gambar 3. 31 *Query* pengecekan tipe promo

Gambar 3.31 adalah *query* yang digunakan untuk melakukan pengecekan promo yang sedang berlangsung, produk mana yang memiliki promo tersebut, dan mengkategorikan jenis promo.



Gambar 3. 32 Hasil pengecekan *query* tipe promo

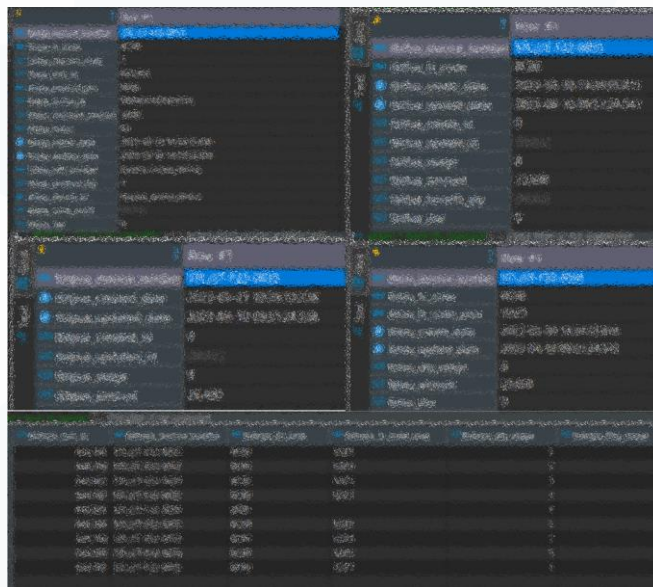


Gambar 3.32 adalah hasil dari *query* pengecekan tipe promo yang digunakan agar *quality assurance* dapat memudahkan untuk melakukan pemisahan tipe promo dan membantu melakukan *testing* skenario didalam bagian klasifikasi dan *test* tipe promo.



Gambar 3. 33 *Query* pengecekan usage promo

Gambar 3.33 adalah gambar pembuatan *query* untuk melakukan pengecekan penggunaan promo dan pendataan pembelian menggunakan promo di aplikasi.



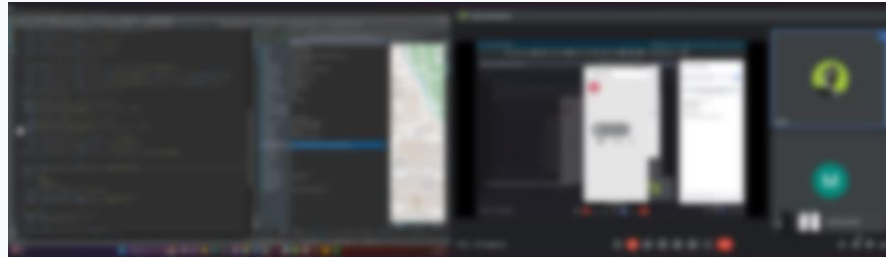
Gambar 3. 34 Hasil *query* pengecekan usage promo

Gambar 3.34 adalah gambar hasil *query* pengecekan *user* promo yang membantu *quality assurance* untuk mendata apakah data promo yang digunakan untuk transaksi tersebut sudah sesuai dan pemakaian sesuai dengan jumlah promo yang digunakan.

### 3.2.3.7 Melakukan *enhancement* fitur

Pada saat melakukan pengetesan di halaman AMS terdapat salah satu ui yang menurut mahasiswa tidak merepresentasikan atau

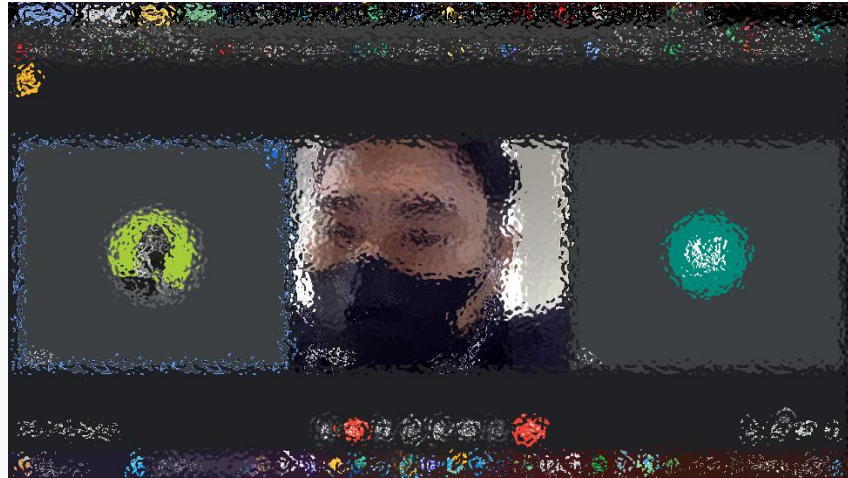
kurang merepresentasikan fungsi sebenarnya karena UI/UX menampilkan hal static. Hal ini dirasa dapat membuat *officer* yang menggunakan bingung apakah tombol dapat ditekan atau telah berkerja karena tidak ada *shadow* atau *animation* sebagai indikasi tombol telah tertekan.



Gambar 3. 35 Rapat dengan *developer* untuk berdiskusi mengenai enhancement

Saat ditemukan potensi enhancement dan perbaikan ini *Quality Assurance* melaporan penemuan ke *manager* dan supervisor divisi prodev yang menyarankan untuk melakukan diskusi terlebih dahulu dengan *developer* yang ditunjukkan pada gambar 3.35 untuk mencari solusi untuk memperbaiki UI dengan asset UI/UX yang sudah ada.

Setelah selesai melakukan rapat untuk membahas mengenai enancement dan potensi perbaikan yang dapat dilakukan untuk asset yang sudah ada tidak dapat ada perbaikan yang dilakukan karena palet warna asset yang ada bentrok dengan warna yang ditentukan untuk button *refresh*, oleh karena itu mahasiswa berbicara dengan tim UI/UX untuk penambahan asset perbaikan UI/UX.



Gambar 3. 36 Rapat Dev, QA, dan UI/UX mengenai enhancement

Gambar 3.36 adalah gambaran meeting yang dilakukan setelah tim UI/UX selesai melakukan penambahan asset dilakukan untuk membahas mengenai hasil desain yang dikerjakan UI/UX dengan *developer* dan *Quality Assurance* untuk memastikan UI yang dibuat sesuai dengan enhancement yang direncanakan dan dapat diimplementasikan ke dalam system. Setelah rapat selesai *developer* mengerjakan perubahan ui dan di *test* kembali oleh *Quality Assurance*.

### 3.2.3.8 Melakukan *full lifecycle testing*

Setelah keseleruhan *bug*, enhancement, dan *test* skenario selesai dilakukan pengetesan terakhir dan final dari *project* yaitu *full lifecycle testing*. *Testing* ini akan mengetes seluruh skenario pembayaran QRIS saja yang dikerjakan dari awal sampai akhir dari sisi Aksesmu dan management *system* dan *database* untuk memastikan seluruh fitur, seluruh data, seluruh aplikasi yang berkaitan dengan *project* atau terkait dengan *project* termasuk promo, permintaan dari vendor dan ke vendor berjalan sesuai dengan ekspektasi. Setelah semua *test* skenario dan *database* selesai dilakukan pengecekan ulang maka *project* dinyatakan selesai dan QA Pass.

### 3.2.3.9 Membuat QA *Documentation project QRIS*



Gambar 3. 37 Dokumentasi *project QRIS*

Setelah *project* dinyatakan selesai, dilakukan pembuatan dokumentasi dari *project QRIS* untuk mencatat seluruh kegiatan pengetestan *project* dan dokumentasi pengetestan *project*. Gambar 3.37 merupakan gambaran Dokumentasi *project QRIS* yang mencakup pencatatan *business rules*, toko yang dipakai, *flow positive* dan *flow negative* dari *project*, pencatatan pengiriman data ke vendor, dan *query* untuk melakukan pengecekan

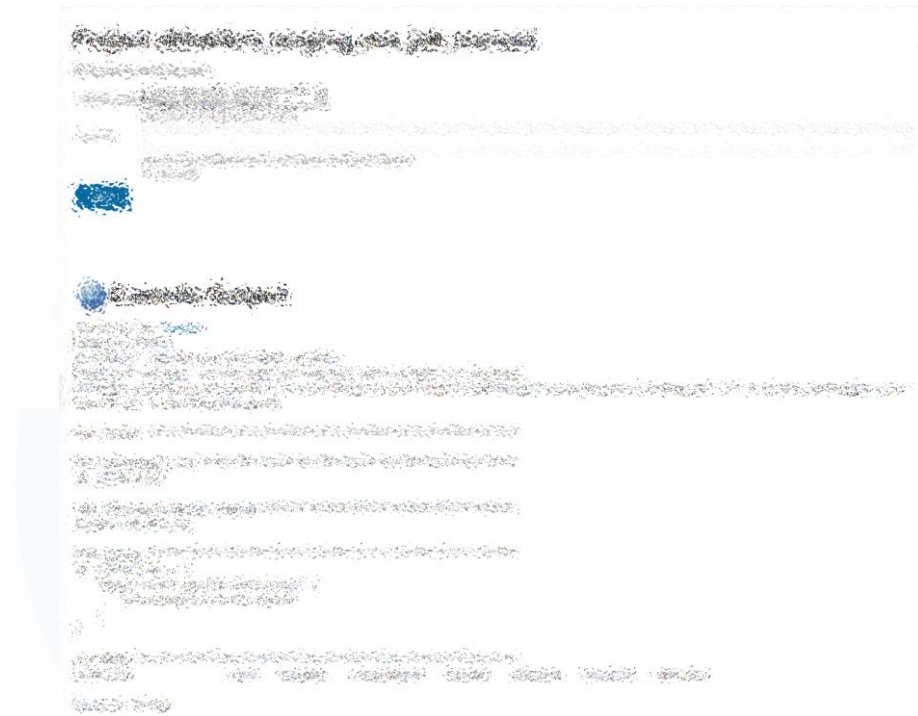
### 3.2.3.10 Melakukan UAT dengan vendor pengiriman



Gambar 3. 38 UAT dengan vendor pengiriman

PT Sumber Trijaya Lestari (Aksesmu) berkerja sama dengan Vendor pengiriman untuk melakukan beberapa pengiriman barang ke *customer*. Gambar 3.38 merupakan gambaran dilakukannya UAT dengan vendor pengiriman untuk melakukan pengecekan data QR yang dikirim dan tampilan sesuai dengan *system back office* aksesmu. *Quality Assurance* bersama tim *product development* dan tim *developer* melakukan QA bersama dari sisi staging dengan tim Vendor pengiriman untuk memastikan data terkirim dengan benar sesuai dengan API dari sisi vendor pengiriman.

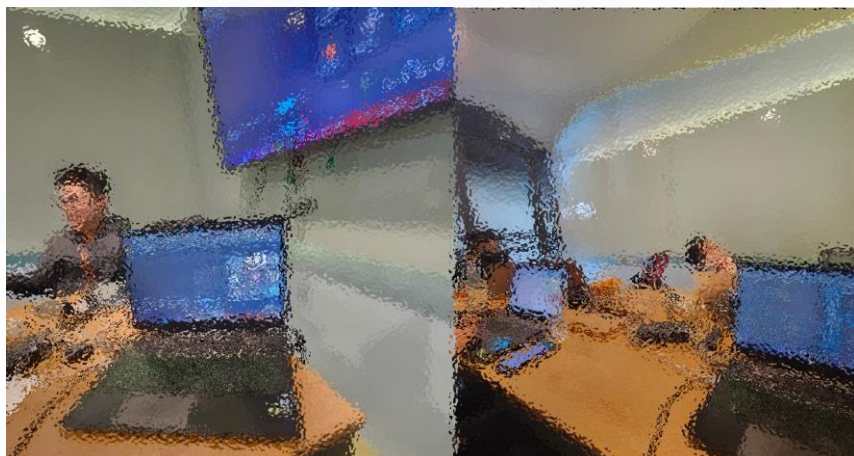




Gambar 3. 39 Jenkins yang dipush untuk Vendor pengiriman

Pengiriman *Job* dilakukan menggunakan Jenkins yang akan mengirimkan data ke *database* vendor pengiriman. API yang dipakai merupakan *Job pengiriman data*, yang akan mengirimkan data sesuai dengan ketentuan aplikasi seperti pada gambar 3.39.

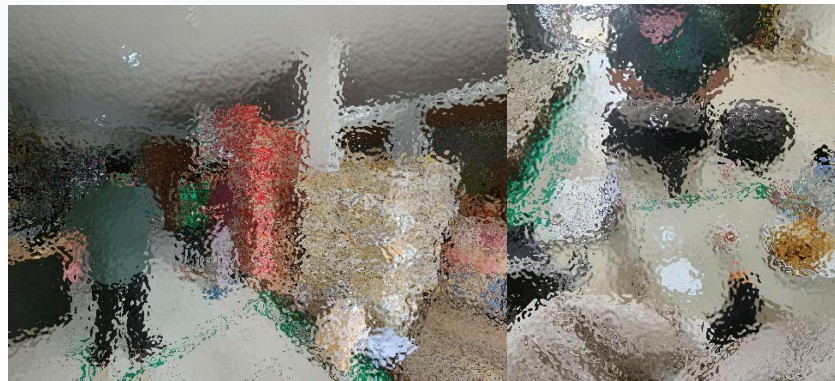
#### 3.2.3.11 Melakukan UAT dengan Operation Manager



Gambar 3. 40 UAT Dengan OM

*Project* yang sudah selesai akan dipresentasikan kepada *user* utama yaitu *operation manager* beserta jajaran-jajaran. UAT dilakukan dengan mempresentasikan flow pembayaran QRIS dari awal mula pesanan dibuat sampai dengan pesanan selesai Gambar 3.40. Dijelaskan seluruh *flow* pembelian dari aplikasi *front office* mulai dari pemilihan pembayaran baru tampilan. Pada saat rapat dilakukan juga tanya jawab dan diskusi mengenai fitur maupun skenario lapangan yang mungkin terjadi ketika melakukan pembayaran QRIS sebagai catatan penambahan untuk enhancement dari *project*.

#### 3.2.3.12 Melakukan TO internal

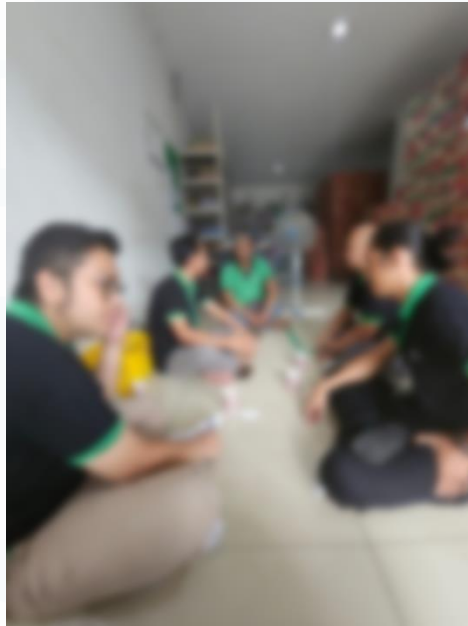


Gambar 3. 41 *Test operation QRIS* pertama

Sebelum melakukan UAT lebih lanjut dengan vendor lain tim aksesmu (*developer, product support specialist, dan Quality Assurance*) berkunjung ke *stock point* untuk melakukan percobaan internal. Tujuan dari pengetesan ini adalah mencoba melakukan pembayaran secara live di toko menggunakan data *production* namun dalam fase beta. Hal ini dikarenakan ada beberapa pengetesan yang tidak dapat dilakukan di staging yang harus dilakukan pengetesan secara live di toko.

Pengetesan ke toko dilakukan selama 2 kali untuk yang pertama bersama dengan tim IT *Operation* dan *developer* pembayaran QRIS. Pada pengetesan pertama ini dilakukan percobaan pembuatan *order*

dan pembayaran dari toko pada saat pengetesan ditemukan beberapa masalah yang terjadi karena kesalahan implementasi dari staging ke beta yang akan diperbaiki setelah kunjungan ditunjukkan pada gambar 3.41.



Gambar 3. 42 *Test operation QRIS kedua*

Pengetesan kedua dilakukan pada tanggal 30 Mei 2023 bersama *superivsor product development, operational manager, Product Support Specialist, dan Quality Assurance*. Pada pengetesan kedua ini tidak ditemukan permasalahan didalam melakukan pembayaran ditunjukkan pada gamabr 3.42.

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



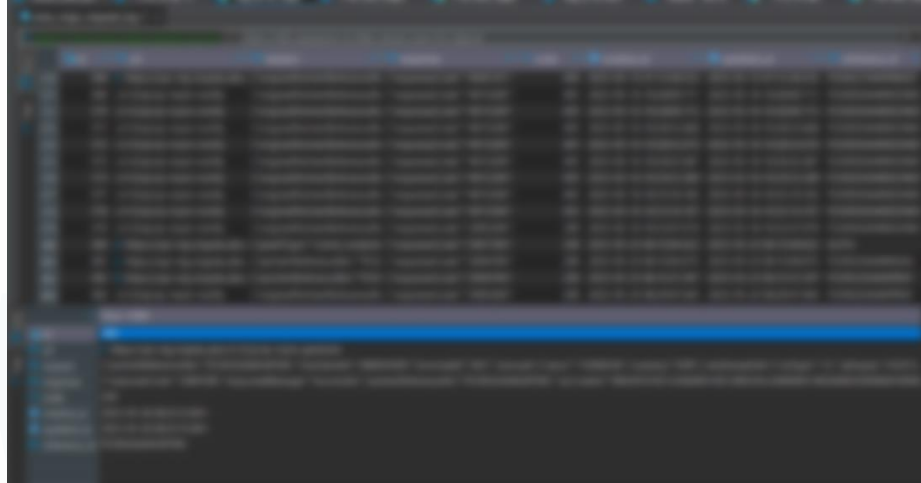
### 3.2.4 Side Project UAT Vendor pembayaran



Gambar 3. 43 Test Fungsionalitas BI

*Project User Acceptance testing* vendor pembayaran adalah *side project* lanjutan dari *project QRIS* yang sudah dilakukan sebelumnya. PT Sumber Trijaya Lestari berkerja sama dengan vendor pembayaran sebagai penghubung dengan Bank Indonesia. Tujuan UAT ini ditujukan agar PT Sumber Trijaya Lestari dapat menyesuaikan validasi *payload* menggunakan *service* yang *didevelop* oleh kedua perusahaan dengan ketentuan *standard operating procedure* (SOP) dalam transaksi QRIS.

Gambar 3.43 adalah contoh pengerjaan skenario yang untuk UAT Vendor pembayaran.



Gambar 3. 44 log *payload service database*

Untuk pengerjaan *project* ini mahasiswa melihat log request di response yang terdapat di dalam *database* postgre dengan aplikasi DBeaver ketika melakukan transaksi dan melakukan pencatatan isi *payload* request dan response di dalam document UAT vendor pembayaran ditunjukkan pada gambar 3.44. Mahasiswa juga berkerja sama dengan *developer enhancement* log QRIS untuk *service case* yang tidak dapat dimunculkan di LOG. Payload yang muncul di log akan dibandingkan dengan column hasil yang diharapkan di document UAT untuk mengetahui apakah request dan response yang diberikan sudah sesuai dengan apa yang diharapkan.

### 3.2.5 **Project Kunjungan AC/AM**

#### 3.2.5.1 Identifikasi *Project Requirement*

*Project* kunjungan AC/AM adalah *project* yang diperuntukan untuk *system* aplikasi *back office* aksesmu management *system* (AMS) untuk melakukan pemberian tugas dan pengawasan kegiatan para area *manager* (AM) dan Area Coordinator (AC).



Gambar 3. 45 business requirement kunjungan AC/AM

Yang pertama dilakukan dalam *project* adalah melakukan pembacaan dan identifikasi *system design* yang diberikan oleh *system analyst* untuk mengetahui *background* dan requirement serta flow dari *project*. Gambar 3.45 merupakan gambaran dari *system design* project kunjungan AC/AM.

*System design* menjelaskan terdapat 3 tabel baru yang akan diimplementasikan untuk melakukan penyimpanan data dari kunjungan AC/AM. Selain itu dapat diketahui *wireframe* dan UI/UX yang dibuat dan limitasi dari sistem serta restriksi yang diterapkan didalam pengembangan fitur untuk setiap menu dan flow *activity diagram* dari setiap *business requirement*.

### 3.2.5.2 Pembuatan *Test Scenario* dan *database mapping*

Setelah melakukan pembelajaran dan analisa mendalam dari *business requirement* yang dilakukan sebelumnya, *user* akan membuat *test* skenario untuk setiap *role* dan *business requirement*.

No	Step	Expected Result	Actual Result
1	Log in as Area Coordinator	Successfully logged in	Successfully logged in
2	Click on 'Add New'	Redirects to the form page	Redirects to the form page
3	Fill in the form fields	Form is saved	Form is saved
4	Click on 'Save'	Redirects to the list page	Redirects to the list page
5	Verify the data is added	Data is visible in the list	Data is visible in the list

Gambar 3. 46 Test skenario bagian satu

Gambar 3.46 merupakan gambaran pembuatan *test* skenario dan database mapping dari skenario bagian satu yang mencakup skenario fitur dan restriksi untuk *role* Area Coordinator.

No	Step	Expected Result	Actual Result
1	Log in as Area Manager	Successfully logged in	Successfully logged in
2	Click on 'Add New'	Redirects to the form page	Redirects to the form page
3	Fill in the form fields	Form is saved	Form is saved
4	Click on 'Save'	Redirects to the list page	Redirects to the list page
5	Verify the data is added	Data is visible in the list	Data is visible in the list

Gambar 3. 47 Test skenario bagian dua

Gambar 3.47 merupakan gambaran pembuatan *test* skenario dan database mapping dari skenario bagian satu yang mencakup skenario fitur dan restriksi untuk *role* Area manager.

No	Step	Expected Result	Actual Result
1	Log in as Area Manager	Successfully logged in	Successfully logged in
2	Click on 'Add New'	Redirects to the form page	Redirects to the form page
3	Fill in the form fields	Form is saved	Form is saved
4	Click on 'Save'	Redirects to the list page	Redirects to the list page
5	Verify the data is added	Data is visible in the list	Data is visible in the list

Gambar 3. 48 Test skenario bagian 3

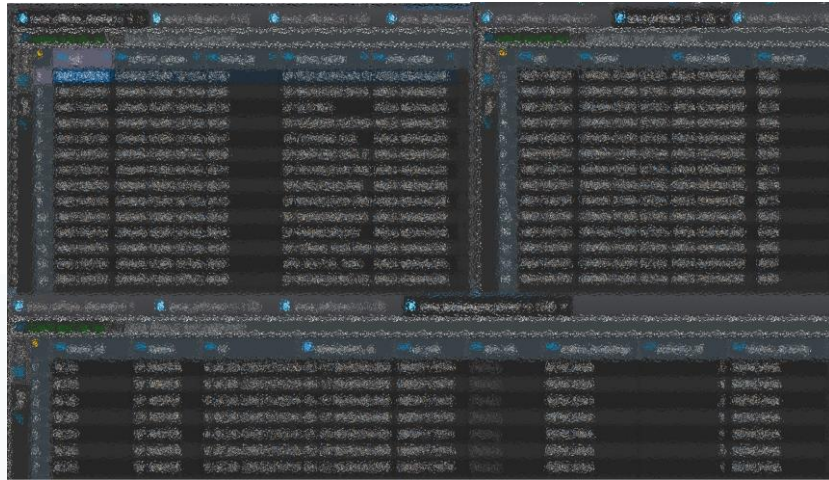
Gambar 3.48 merupakan gambaran pembuatan *test* skenario dan database mapping dari skenario bagian ketiga yang mencakup skenario fitur dan restriksi untuk fitur X.



Gambar 3. 49 *Query* Kunjungan AC/AM

Saat pembuatan *test* skenario selesai dibuat dilakukan *database mapping* untuk membuat pemetaan data yang akan dipakai atau berubah didalam *test* skenario. Pembuatan *database mapping* dilakukan dengan bantuan DBeaver untuk mencatat *database* mana yang dipakai, tabel apa yang akan dipakai, dan column apa yang akan terisi atau berubah. Pada saat melakukan *database mapping* ini juga dibuat *query* seperti pada gambar 3.49.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3. 50 Hasil *Query* kunjungan AC/AM

Gambar 3.50 adalah hasil dari pembuatan *query* untuk untuk kunjungan ac/am yang dibuat untuk melakukan pengecekan kunjungan hari itu, fitur riwayat, dan batasan yang ada di *system design*. Hasil *query* ini juga membantu *quality assurance* untuk melihat perubahan data dari setiap *test* skenario yang dibuat.

### 3.2.5.3 Pengetesan Awal Aplikasi

No	Tgl	Waktu	Lokasi	Status
1	2023-10-26	08:00	Ruang 101	Selesai
2	2023-10-26	09:00	Ruang 102	Selesai
3	2023-10-26	10:00	Ruang 103	Selesai
4	2023-10-26	11:00	Ruang 104	Selesai
5	2023-10-26	12:00	Ruang 105	Selesai
6	2023-10-26	13:00	Ruang 106	Selesai
7	2023-10-26	14:00	Ruang 107	Selesai
8	2023-10-26	15:00	Ruang 108	Selesai
9	2023-10-26	16:00	Ruang 109	Selesai
10	2023-10-26	17:00	Ruang 110	Selesai

Gambar 3. 51 Sheet pengetesan awal rencana kunjungan AC

Karena aplikasi pada saat dibuat *test* skenario dan *database mapping* berada di tahap *development*, fitur yang dilakukan pengetesan adalah fitur awal untuk kunjungan AC/AM. Dari pengetesan awal dapat terlihat bagaimana data akan tersimpan didalam *database* dan dapat dilakukan pembuatan *query* lanjutan untuk data tersebut. Gambar 3.51 adalah gambaran pengetesan awal untuk rencana kunjungan AC.

No.	Uraian	Status	Penyakit	Penyakit	Penyakit
1	...	...	...	...	...
2	...	...	...	...	...
3	...	...	...	...	...
4	...	...	...	...	...
5	...	...	...	...	...

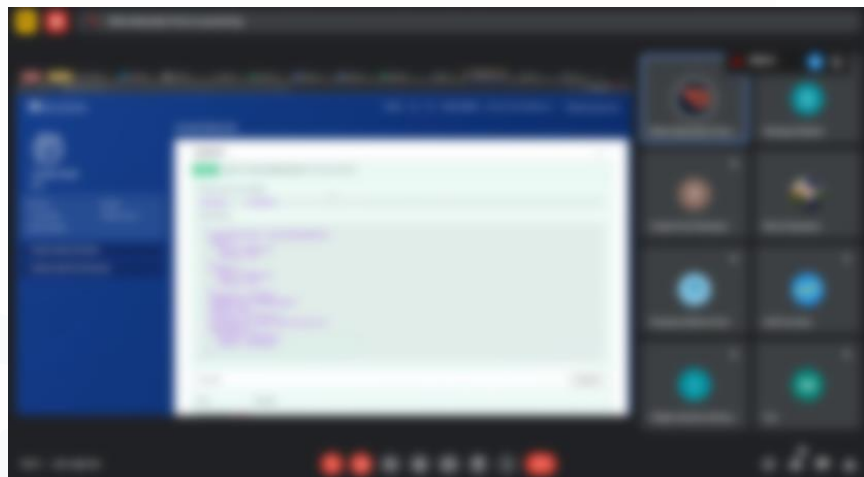
Gambar 3. 52 Contoh pelaporan *bug* rencana kunjungan AC

Segala *bug* yang ditemukan tetap akan tetap dicatat di *bug list* sebagai catatan ke *developer* dalam melakukan pembuatan aplikasi. Gambar 3.52 adalah sheet pelaporan *bug* awal untuk rencana kunjungan AC.

### 3.2.6 Side project tes fungsionalitas SNAP Bank Indonesia

#### 3.2.6.1 Training Coaching Clinic SNAP

*Project* tes fungsionalitas Bank Indonesia adalah *project* lanjutan dari *project* QRIS. Output dari *project* tes fungsionalitas akan diberikan ke Bank Indonesia yang membuktikan bahwa *system* pembayaran QRIS telah mengikuti Standar Nasional *Open API* (SNAP). SNAP memastikan bahwa QRIS dari sisi client aksesmu memiliki protokol dan instruksi yang sama sebagai jembatan koneksi antara satu aplikasi dengan aplikasi yang lain.



Gambar 3. 53 Rapat Dengan Vendor pembayaran Untuk SNAP

Diadakan rapat dengan vendor pembayaran mengenai *Coaching Clinic SNAP* dan apa yang harus dikerjakan sebagai autentikasi Aksesmu ke Bank Indonesia. Mahasiswa mendapatkan bagian



mengerjakan tes fungsionalitas untuk melakukan pengecekan service dan *payload* untuk melihat apakah *payload* sudah sesuai standar Bank Indonesia. Gambar 3.53 adalah sesi *coaching clinic* SNAP yang dilakukan dengan vendor pembayaran.

### 3.2.6.2 Pengerjaan *Test* Fungsionalitas



No	Uji	Kasus	Skenario	Detail	Hasil	Catatan	Tgl
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

Gambar 3. 54 Document *Test* Fungsionalitas SNAP

Pengerjaan dilakukan bersama dengan *developer* untuk mengisi *testing* document karena permintaan *service* Bank Indonesia yang mengharuskan mencatumkan *confidential data* dari setiap *service* yang didokumentasikan. Terdapat beberapa permasalahan yang muncul pada saat pengerjaan, seperti susah mengambil *header request* yang terdapat di API, salahnya *payload* yang diberikan dari sisi vendor pembayaran maupun dari aksesmu yang membuat ada beberapa perbaikan didalam *service* yang digunakan. Gambar 3.54 adalah *sheet* hasil dari pekerjaan melakukan tes fungsionalitas SNAP.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 3.2.7 Project Enhancement Setoran

#### 3.2.7.1 Rapat dengan tim untuk identifikasi *project*



Gambar 3. 55 Rapat untuk *Project* Setoran

Pada awal *project* dilakukan sebuah rapat dengan tim prodev dan *developer* mengenai enhancement *project* baru. Didalam rapat dijelaskan mengenai *background* dari *project*, ekspetasi dari *project*, pembagian tugas dari tim, *database* berserta column yang akan dipakai, UI/UX dan flow dari aplikasi. *Project* ini memiliki status Very Important *Project* yang harus diselesaikan secepat mungkin yang membuat *development project* kunjungan AC/AM di hold untuk sementara waktu. Gambar 3.55 merupakan gambaran dilakukannya rapat untuk *project* setoran.



Gambar 3. 56 Rangkuman *project requirement* dan *system design project* setoran

*Project* setoran dibuat untuk mengakomodasi *officer* toko yang sudah mengirimkan barang ke *customer* dengan metode bayar di tempat dan ingin melakukan setoran. *Enhancement* yang dilakukan adalah penambahan tipe setoran yang sebelumnya hanya dapat dilakukan ke Manajer Toko menjadi dapat dilakukan juga di gerai-gerai Alfamart dengan harapan dapat menambah opsi setoran di perusahaan memudahkan pegawai dalam melakukan setoran ke perusahaan. Gambar 3.56 adalah rangkuman dari *project requirement* dan *system design* yang dipaparkan selama rapat.

### 3.2.7.2 Pembuatan *Scenario* dan *database mapping*



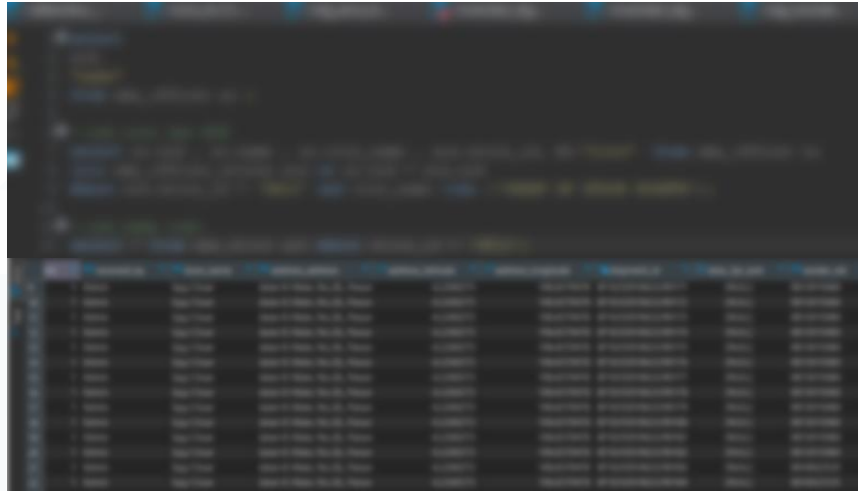
Gambar 3. 57 UI/UX setoran

*Project* ini dikategorikan sebagai proyek penting, familiarisasi UI/UX, *database*, dan pembacaan *system design* lebih lanjut dilakukan secara bersamaan dengan pembuatan *test* skenario. Pembuatan skenario dibagi menjadi dua bagian karyawan pengiriman dan *manager* toko dimana setiap bagian dikerjakan oleh satu *Quality Assurance*. Gambar 3.57 merupakan mockup UI/UX yang dibuat untuk *project* setoran.

No	Uraian	Langkah	Data	Diharapkan	Actual	Pass/Fail	Catatan
<b>Test Scenario</b>							
1	Penyimpanan	1. Klik menu 'Setor' di bagian atas layar.		Muncul layar pilihan metode setoran.			
2	Penyimpanan	2. Pilih metode setoran 'via Alfamart'.		Muncul layar konfirmasi setoran via Alfamart.			
3	Penyimpanan	3. Pilih metode setoran 'via Manajer'.		Muncul layar konfirmasi setoran via Manajer.			
4	Penyimpanan	4. Masukkan nominal setoran.	10000	Muncul layar konfirmasi nominal setoran.			
5	Penyimpanan	5. Klik tombol 'Setor'.		Muncul layar konfirmasi akhir setoran.			
<b>Test Data</b>							
1	Penyimpanan	1. Klik menu 'Setor' di bagian atas layar.		Muncul layar pilihan metode setoran.			
2	Penyimpanan	2. Pilih metode setoran 'via Alfamart'.		Muncul layar konfirmasi setoran via Alfamart.			
3	Penyimpanan	3. Pilih metode setoran 'via Manajer'.		Muncul layar konfirmasi setoran via Manajer.			
4	Penyimpanan	4. Masukkan nominal setoran.	10000	Muncul layar konfirmasi nominal setoran.			
5	Penyimpanan	5. Klik tombol 'Setor'.		Muncul layar konfirmasi akhir setoran.			

Gambar 3. 58 *Test* Skenario Setoran

Gambar 3.58 adalah Pembuatan sheet dari sisi karyawan pengiriman, *Test* skenario dibagi menjadi 2 bagian. Bagian setoran via Alfamart dan via Manajer



Gambar 3. 59 Pembuatan dan hasil *Query project* setoran

Dari dua bagian skenario yang dibuat dilakukan *database exploration* dan *mapping* untuk setiap *test* skenario yang menggunakan atau menampilkan data dengan bantuan aplikasi Dbeaver. Selain itu dibuat *query* untuk memudahkan pengecekan data dari tabel dan kolom yang berbeda. Gambar 3.59 adalah *query* yang dibuat untuk melakukan pengecekan *database*, sedangkan gambar 3.63 adalah gambaran *query project* setoran. Selain itu dilakukan eksplorasi API untuk melakukan pengecekan API yang terkait dengan setoran.

### 3.3 Kendala yang Ditemukan

Terdapat beberapa kendala yang ditemukan dalam proses pengetesan beberapa *project* di PT Sumber Trijaya Lestari, diantaranya:

1) Kesulitan dalam mengidentifikasi *database* perusahaan

PT Sumber Trijaya Lestari memiliki banyak *database* yang terdiri dari beberapa tabel. Rumitnya penamaan dan banyaknya data membuat *database* sulit untuk dipahami, sehingga dibutuhkan waktu yang cukup lama untuk bisa mengidentifikasi serta memahami setiap *database* di perusahaan.

2) Kesulitan dalam beradaptasi dengan *flow business* perusahaan.

PT Sumber Trijaya Lestari memiliki banyak aplikasi untuk menunjang operasional perusahaan. Penjelasan oleh supervisor mengenai masing-masing aplikasi beserta dokumen *testing* tidak cukup menjelaskan keseluruhan dari flow business aplikasi. Oleh karena itu, dibutuhkan waktu yang cukup banyak untuk memahami *flow business* dari masing-masing aplikasi.

### 3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi permasalahan yang muncul di PT Sumber Trijaya Lestari terdapat beberapa solusi yang diusulkan:

- 1) Explorasi *database* lebih dalam dengan mencoba data aplikasi. Salah satu cara cepat untuk dapat mempelajari lebih dalam *database* perusahaan adalah dengan mencoba aplikasi secara langsung dan melihat perubahan data yang terjadi di aplikasi. Selain itu dari dokumentasi-dokumentasi *project-project* yang sudah dilakukan sebelumnya dapat ditemukan *database* yang dipakai dan data apa saja yang dipakai untuk fitur tersebut. Selain itu bertanya kepada tim mengenai perubahan data atau data yang membingungkan dapat mempermudah *Quality Assurance* dalam memahami tipe data yang ada dalam *database*.
- 2) Mencoba langsung aplikasi dan terjun ke lapangan. Flow business dan aplikasi perusahaan yang banyak dapat dipelajari dengan mencoba langsung berbagai *system* dan fitur yang ada di PT Sumber Trijaya Lestari. *Quality Assurance* dalam waktu luang melakukan banyak pembelajaran terhadap *system* lain perusahaan yang tidak dikerjakan dalam *project* untuk memperdalam *business knowledge* dan *understanding* perusahaan. Selain itu terjun ke lapangan dan melihat langsung proses bisnis juga dapat sangat membantu *Quality Assurance* mempelajari hal-hal yang mungkin tidak ada dalam dokumen sebelumnya dan mendapatkan banyak pengetahuan tentang bisnis.