

BAB III

PELAKSANAAN KERJA PROFESI

Pada bab ini akan menjelaskan proses-proses saya melaksanakan kerja profesi.

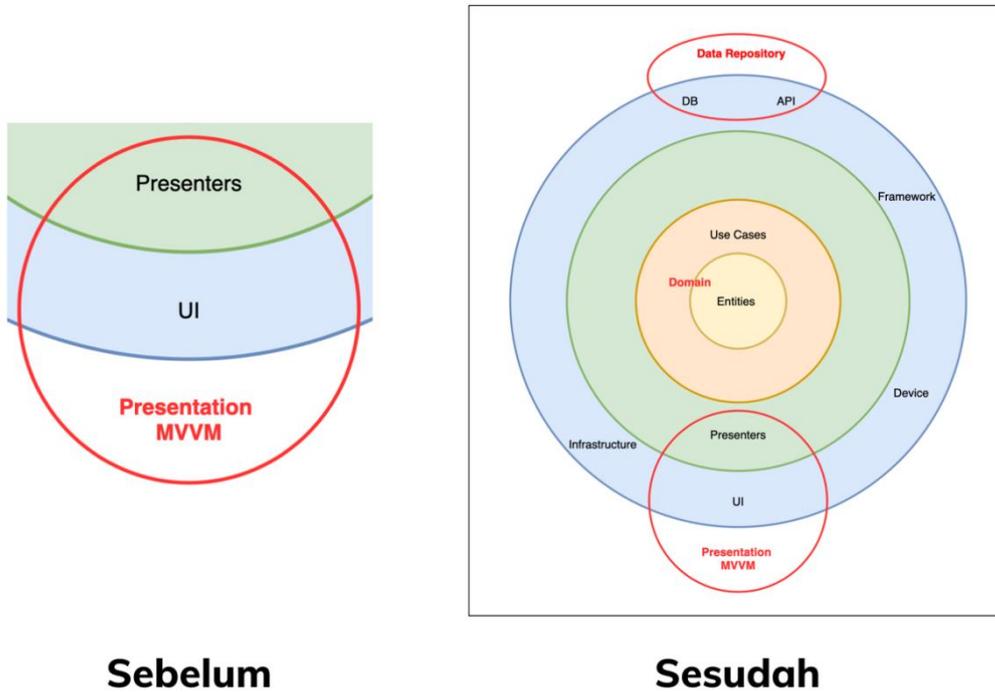
3.1 Bidang Kerja

Saya di PT Fintek Karya Nusantara (Linkaja) ditugaskan oleh atasan saya untuk bergabung ke dalam tim/Squad yang bernama Bedah Rumah, yang bertujuan untuk membuat aplikasi Linkaja baru dengan arsitektur baru yakni *Clean Architecture* yang bertujuan untuk menggantikan aplikasi Linkaja yang beredar dipasaran yang basisnya dari aplikasi T-Cash.

Alasan kenapa aplikasi Linkaja ditulis/dibuat ulang (*Rewrite*) karena Aspek-aspek dalam pengembangan kualitas kode dari aplikasi semakin menurun khususnya aspek *Maintainability* seperti *Responsibility* dalam suatu File sudah tercampur-aduk yang mengakibatkan code *project* membengkak & baris File tersebut menjadi masif dan akan terus mengganggu pengembangan fitur yang berkesinambungan. Ditambah lagi dengan perbedaan antara tim iOS dan Android dalam mendefinisikan suatu fungsi / *function* dari blok suatu *code*, sehingga fitur yang sudah selesai dikerjakan dan mendapatkan *Request* dari *Product Owner* untuk dilakukan improvisasi fitur tersebut, kedua tim iOS dan Android selanjutnya akan sulit untuk berkomunikasi untuk melanjutkan fitur tersebut karena perbedaan dalam penyebutan/mendefinisikan suatu *Function, Object, Variable*.

Oleh karena itu menjadi alasan kami memutuskan untuk membuat aplikasi Linkaja baru dengan arsitektur baru yaitu *Clean Architecture*, yang bertujuan untuk memisahkan *Layer-layer* dalam struktur pengembangan aplikasi, dan tetap mempertahankan *Model-View-ViewModel* sebagai *Design Pattern* aplikasi yang termasuk dalam bagian *Presentation Layer*.

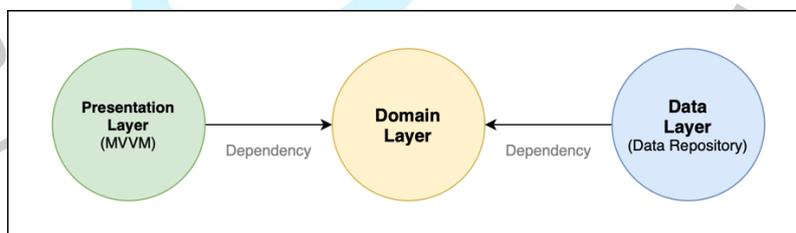
3.1.1 Layer-layer pada Clean Architecture Linkaja



Gambar 3.1 Rearchitecture Linkaja ke Clean Architecture
 Dokumentasi Linkaja Teknologi (2020)

Berikut ini adalah hasil pengelompokan masing-masing Layer:

Presentation, Domain, dan Data Layer.



Gambar 3.2 Pengelompokan Layer-layer pada Arsitektur
 Dokumentasi Linkaja Teknologi (2020)

a. Presentation Layer

Berisi *UI* (UIViewControllers atau SwiftUI). Tampilan dikoordinasikan oleh *ViewModel (Presenter)* yang menjalankan satu atau banyak *use case*. *Presentation Layer* hanya bergantung pada *Domain Layer*.

b. Domain Layer

Adalah bagian paling dalam arsitektur ini (tanpa ketergantungan ke layer lain, layer ini benar-benar terisolasi). Pada domain layer berisi Entity (Business Model), Use Case, and Repository Interface. Layer ini berpotensi digunakan kembali dalam berbagai proyek.

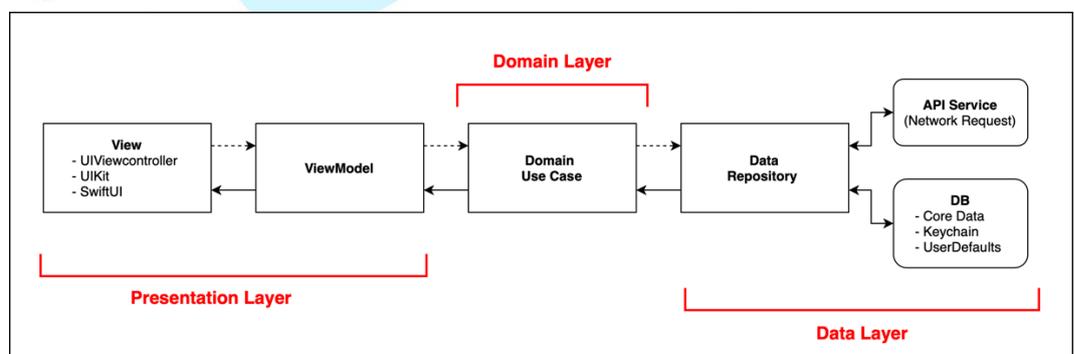
Pemisahan tersebut memungkinkan untuk tidak menggunakan aplikasi host dalam target testing karena tidak ada ketergantungan (dengan Third-Party Library) yang diperlukan - hal ini membuat testing pada Domain use case akan hanya perlu waktu yang singkat.

Catatan: Domain layer tidak boleh menyertakan apa pun dari layer lain (misal. Presentation Layer - UIKit atau SwiftUI atau Data Layer - Mapping Codable)

c. Data Layer

Berisi penerapan *Repository* dan satu atau banyak *Data Source*. Repository bertanggung jawab untuk mengoordinasikan data dari Data Source yang berbeda. Data Source bisa Remote atau Local. Data Layer hanya bergantung pada Domain Layer. Pada layer ini, kita juga dapat menambahkan *mapping network* untuk *JSON Data Decodable* dan *Entity Local Database*.

3.1.2 Data Flow pada Clean Architecture Linkaja



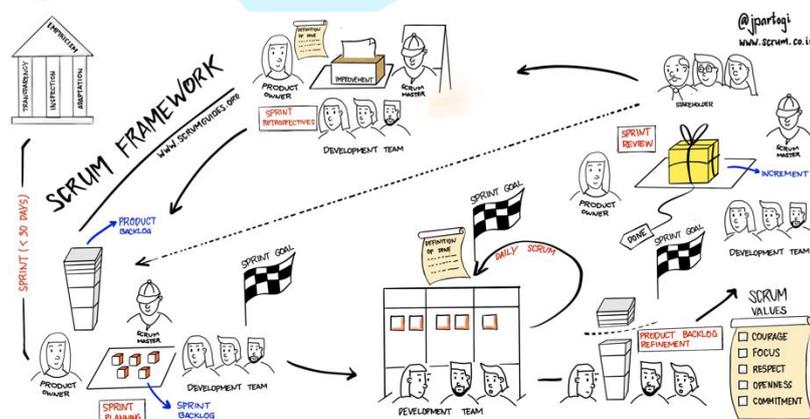
Gambar 3.3 Detail Alur Data (*Data Flow*) pada Arsitektur Dokumentasi Linkaja Teknologi (2020)

Data Flow ini menjelaskan alur proses dari masing-masing layer, sebagai berikut:

1. **View (UI)** akan memanggil method dari **ViewModel (Presenter)**.
2. **ViewModel** akan mengeksekusi **Use Case**.
3. **Use Case** sesuai fungsinya akan mengeksekusi **Repository**.
4. Pada setiap **Repository** akan mengembalikan data dari *Remote Data (Network)*, *Local Data (Database, User Default* atau pun dari *Keychain*).
5. Informasi ini akan dikembalikan ke **View (UI)** dimana data itu seharusnya ditampilkan.

3.2 Pelaksanaan Kerja

Pelaksanaan kerja praktik dalam mengembangkan aplikasi Linkaja yang berkesinambungan mengacu pada metode *Software Development Life Cycle Scrum*, sederhananya *Scrum* itu pengembangan produk yang dilakukan secara iteratif dan inkremental. Dan disini kami memiliki periode kerja (*Sprint* dalam definisi *Scrum*) setiap 2 pekan untuk mengerjakan daftar *Task*/pekerjaan modul-modul yang harus diselesaikan. Setelah periode itu selesai maka akan lanjut dengan periode selanjutnya dengan daftar *task*/pekerjaan yang baru yang telah diputuskan dari *Sprint Planning* dengan daftar *Product Backlog* (daftar kerjaan yang belum diputuskan untuk diambil atau tidak).

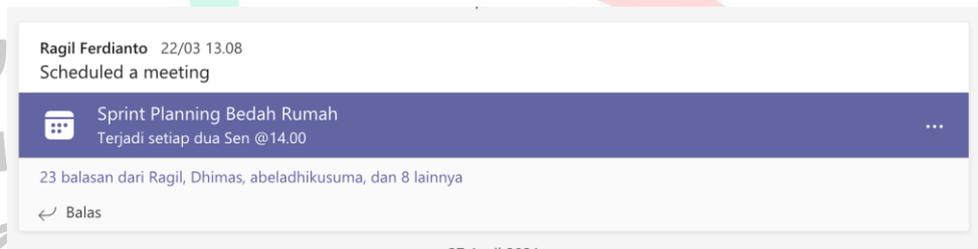


Gambar 3.4 Detail Alur SDLC Scrum
What is Scrum, Scrum.co.id (2020)

Dan saya dipindahkan ke Tim/Squad Bedah Rumah (tim yang bertujuan untuk membuat aplikasi Linkaja baru) yang sedang berjalan, mulai dari periode/*Sprint* ke 7 mulai dari 1 Juni, sampai *Sprint* 12 yang dilanjutkan dengan *Internal Alpha Testing*, *Internal Beta Testing* dan diakhiri dengan perilisan aplikasi aplikasi Linkaja ke pasaran pada 2 *Platform*, yakni iOS dan Android.

3.2.1 *Sprint Planning*

Seperti yang disinggung pada sub-bab sebelumnya, *Sprint Planning* itu menjadi awal dari perencanaan daftar modul-modul/tiket fitur yang akan diambil dan dikerjakan pada periode/*Sprint* selanjutnya. Diselenggarakan setiap hari Senin, 2 pekan sekali secara *Online* melalui *Meeting* di Teams yang dipimpin oleh Scrum Master sebagai penyelenggara dan diikuti sertakan tim Mobile App Developer, Back-End Developer, System Analyst, Product Owner dan UI/UX Designer.



Gambar 3.5 Jadwal Rutin Meeting *Sprint Planning*
Linkaja Bedah Rumah *Squad*, Microsoft Teams (2021)

Di rangkaian acara ini kami membahas dan meninjau bersama-sama dari tiap tiket modul dari *Product Backlog* yang akan diambil ke *Sprint* selanjutnya antara lain meninjau ringkasan, prekondisi, *Business Requirement*, *Acceptance Criteria Testing*. Tiket yang dari prioritas tinggi yang akan dibahas terlebih dahulu. Agenda ini didiskusikan biasanya membutuhkan waktu yang bisa dibilang cukup lama hingga 2jam bahkan lebih. Hasil akhir dari *Meeting* ini ialah menentukan siapa Developer yang mengambil suatu tiket/modul dengan pertimbangan atas kemauan sendiri dan beban kerja dari masing-masing individu yang sudah ada, jadi sama sekali tidak ada paksaan dari pihak manapun.

Projects / Bedah Rumah Squad / Scrum Board

Backlog

05/Apr/21 4:12 PM - 19/Apr/21 4:12 PM

Defect Alpha Defect Beta Only My Issues Recently Updated Insights

Bedah Rumah Sprint 10 473 issues

05/Apr/21 4:12 PM - 19/Apr/21 4:12 PM

Item	Issue ID
Refactoring QRIS Transfer	RA-860
Android - Testing end to end - Peer to Peer	RA-16
Testing end to end - Android- Splash Screen	RA-69
Airtime Pulsa Data - Input ID	RA-134
QR input Amount	RA-312
QR Inquiry	RA-314
QR Parser	RA-315
Recurring Create	RA-289
Recurring List	RA-286
Recurring Detail	RA-288
Recurring Edit	RA-287
Flow Diagram Recurring	RA-291

Gambar 3.6 Daftar Modul Yang Akan Diambil Linkaja Scrum, Jira (2021)

Tidak ada ketentuan tiap Developer harus mengambil berapa tiket/modul dalam satu *Sprint*/periode.

Projects / Bedah Rumah Squad / RA-995 / RA-1012

iOS - Permission - Notification Home

QA PASSED

Attach Link issue

Description
Summary:
 Permission di aplikasi linkaja untuk beberapa akses di device.

Precondition:

- User aplikasi LinkAja basic dan full service
- user adalah user linkaja
- user akses ke aplikasi linkaja

Business Requirement:
 pengerjaan design frontend android dan ios untuk aplikasi linkaja butuh akses atau allow permission (sesuai dengan figma).

Notes:
 test dilakukan hanya mengacu pada figma dengan flow dan service Backend existing. (hanya memvalidasi berdasarkan design di figma)

Acceptance Criteria:

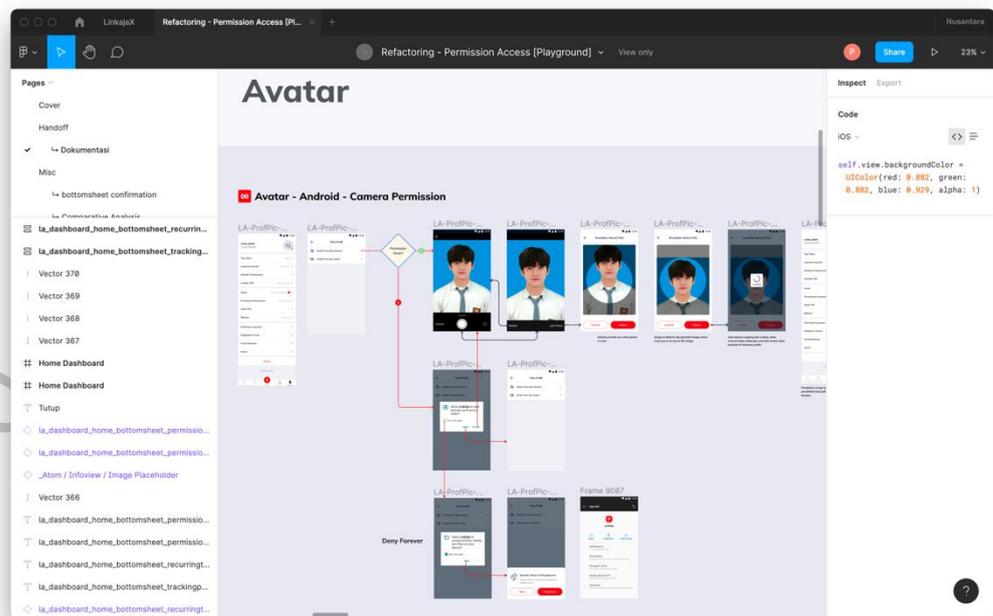
- user install aplikasi linkaja
- user bisa akses fitur

Add a comment...
 Pro tip: press M to comment

Field	Value
Assignee	Pengguna (Dary)
Reporter	Risma Hardiyani
Development	5 commits 4 months ago
Labels	None
Tester	Sri wahyuni
UX Designer	None
Tandem	None
Estimation (Task)	None
Developer	None
Sprint	Bedah Rumah Sprint 10
Priority	Medium
TestRail: Cases	Open TestRail: Cases
TestRail: Runs	Open TestRail: Runs

Gambar 3.7 Informasi Detail Modul Yang Akan Diambil Linkaja Scrum, Jira (2021)

Ketika Developer mengambil tiket akan ditanya lagi apakah beban kerja untuk periode Sprint ini masih menyanggupi dan bisa berkomitmen untuk menyelesaikannya dengan tepat waktu.



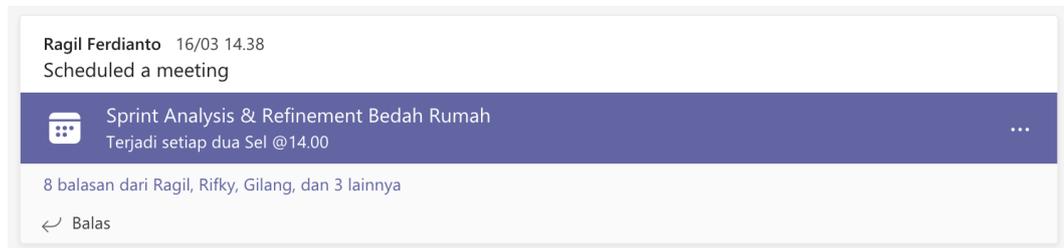
Gambar 3.8 Alur Tampilan Yang Akan Diimplementasi
Linkaja UI Documentation, Figma (2021)

Dan dibahas juga dokumentasi *UI/UX* dari tiap modul/tiket *Product Backlog* yang sedang didiskusikan yang akan diimplementasikan menjadi sebuah *code*. Sering kali terjadi komponen dari suatu *UI (User Interface)* tidak sesuai dengan *Guidelines* dari tiap *Platform Mobile* dari Android maupun iOS, sehingga modul *UI/UX* itu akan direvisi karena beberapa kasus antara lain navigasi dari suatu halaman ke halaman tidak efektif/efisien, tidak menggunakan *Resources* yang sudah ada hingga terdapat tampilan yang belum tersedia/selesai dan banyak lainnya.

3.2.2 Sprint Analysis

Seperti yang saya sampaikan sebelumnya, Berpartisipasi dalam rangkaian acara *Sprint Planning* begitu lama karena kami sebagai Developer membahas suatu tiket yang kita belum ketahui dan terjadi

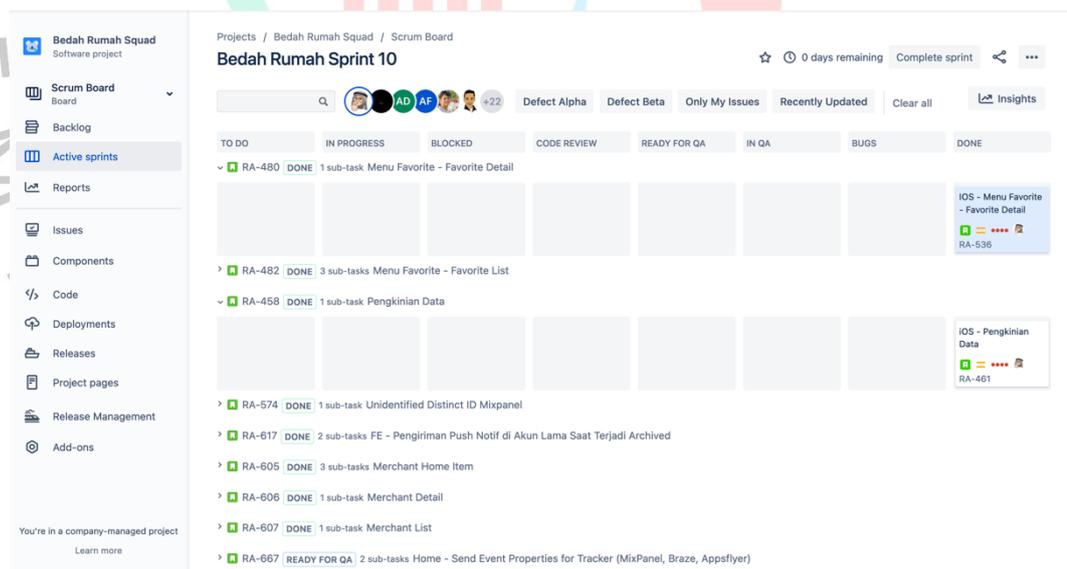
bahasan yang lama di tiap tiket tersebut. Oleh karena itu untuk menghindari waktu diskusi yang lama, terdapat rangkaian acara sebelum *Sprint Planning* bernama *Sprint Analysis*, tujuannya untuk membahas ringkasan aktivitas *Sprint Planning*, sehingga ketika nanti di rangkaian *Sprint Planning*



Gambar 3.9 Jadwal Rutin Meeting Sprint Planning
Linkaja Bedah Rumah Squad, Microsoft Teams (2021)

yang akan datang, para Developer sudah lebih mafhum apa yang akan dibahas nanti dalam tiap item-item tiket/modul yang terdapat dalam *Product Backlog*.

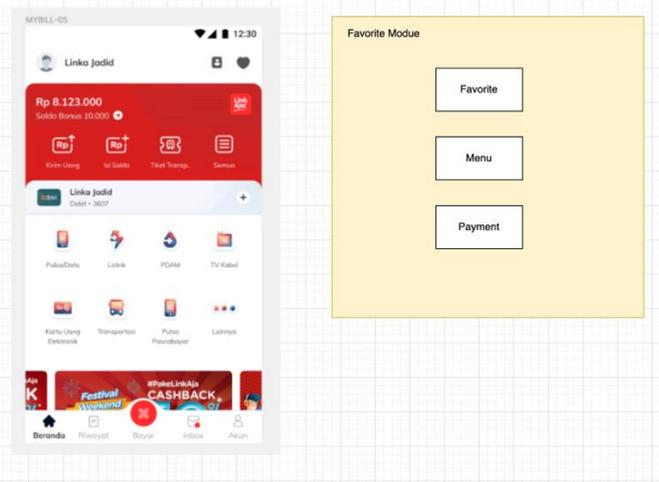
3.2.3 Pra Mulai Coding Pengembangan Modul



Gambar 3.10 Active Sprint/Periode Yang Sedang Jalan
Linkaja Bedah Rumah Squad, Jira (2021)

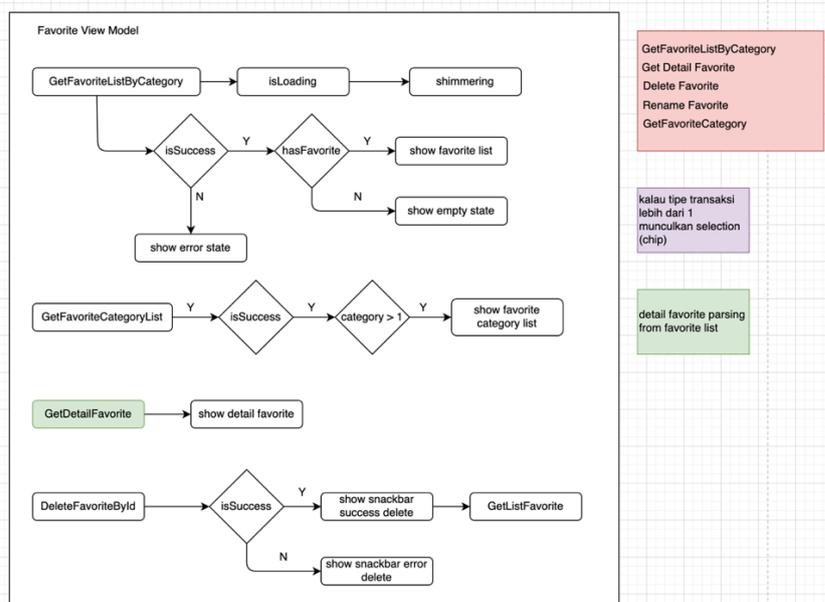
Setelah mengikuti agenda *Sprint Planning* dan telah mengambil tiket-tiket untuk dikerjakan, kita tidak bisa langsung *Coding* mengerjakan modul tiketnya, tapi harus membuat *Diagram Flow* dari modul yang akan

dikerjakan terlebih dahulu bekerja sama tim antar Platform iOS & Android yang mengambil tiket/modul yang sama, dari membuat diagram *Presentation Layer* meliputi *ViewModel*, *Domain Layer* meliputi *UseCase*, hingga *Data Layer* meliputi *Repository* dan *DataSource* yang meliputi *Local Data* dan *Remote Data*.



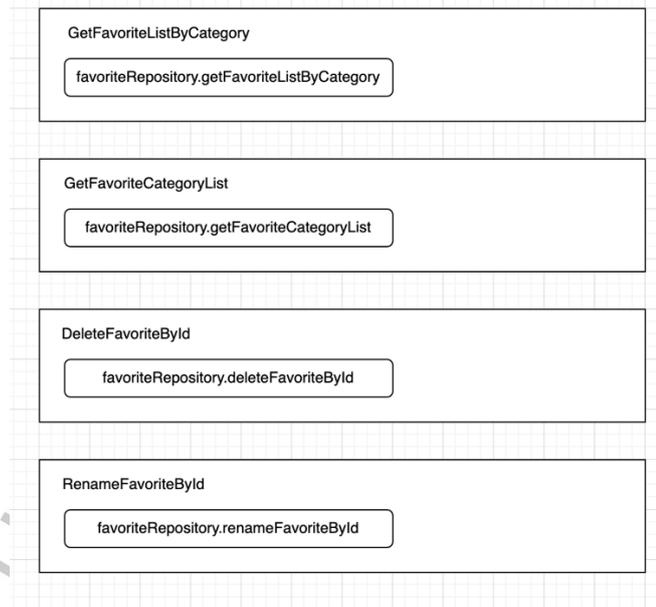
Gambar 3.11 Diagram Flow Modul Favorit Linkaja Bedah Rumah Squad, Draw.io (2021)

Diagram Flow ini menjadi beranda dari diagram modul favorit.



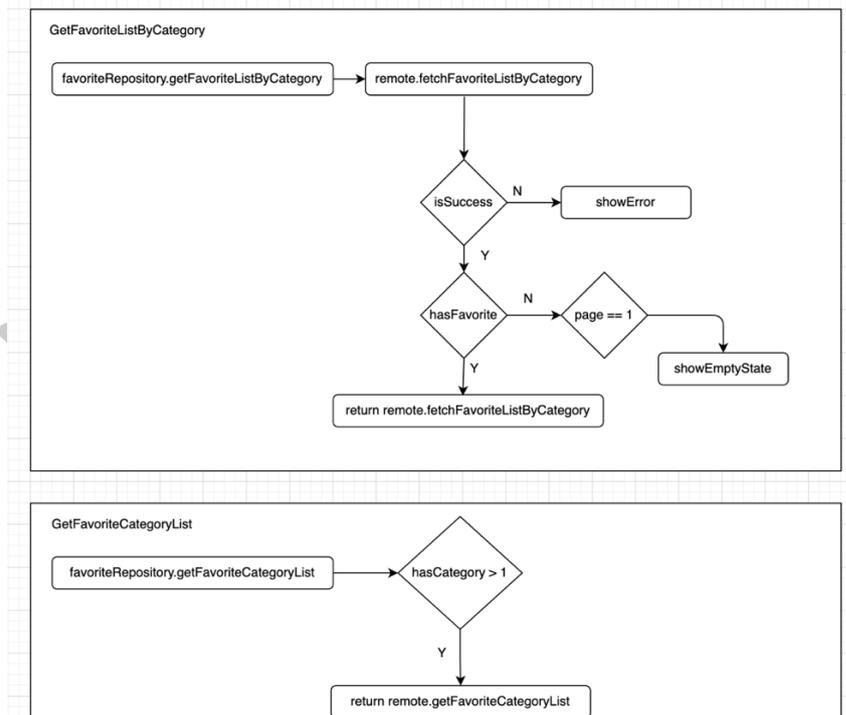
Gambar 3.12 Diagram Flow ViewModel Modul Favorit Linkaja Bedah Rumah Squad, Draw.io (2021)

Membuat *Diagram Flow ViewModel* merupakan langkah selanjutnya.



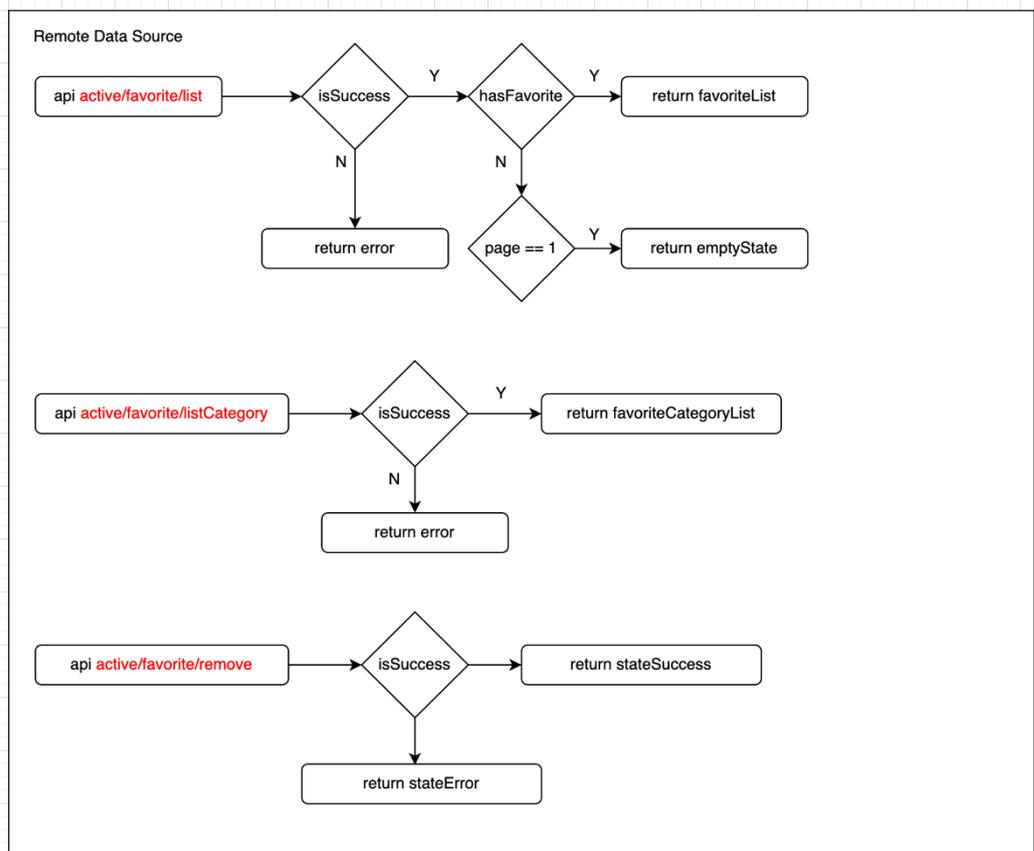
Gambar 3.13 Diagram Flow UseCase Modul Favorit Linkaja Bedah Rumah Squad, Draw.io (2021)

Dilanjutkan dengan membuat *Diagram Flow UseCase*.



Gambar 3.14 Diagram Flow Repository Modul Favorit Linkaja Bedah Rumah Squad, Draw.io (2021)

Dan dilanjutkan dengan *Diagram Flow Repository*.



Gambar 3.15 Diagram Flow DataSource Modul Favorit
Linkaja Bedah Rumah Squad, Draw.io (2021)

Setelah pembuatan *Diagram Flow* untuk suatu modul selesai atas kesepakatan kedua belah pihak yakni tim Android dan iOS, maka dapat dilakukan pengerjaan *Coding* modul tiket tersebut dengan membuat *Git Branch* terlebih dahulu dengan ketentuan alur yang sudah disepakati bersama yang akan dijelaskan pengerjaannya di sub-bab setelah ini.

Saat *Sprint Planning* sudah dimulai, dilakukannya aktivitas *meeting* harian bernama *Sprint Daily Meetup* yang berdurasi 15-30 menit dalam tiap harinya, tujuannya untuk memfasilitasi komunikasi Developer ketika menghadapi masalah dalam pengerjaan tiket modul yang diambil. Sehingga dengan komunikasi disini diharapkan terhindar dari telatnya dalam *deliver* suatu tiket modul/produk. Karena tiap tiket modul yang telat, akan dibawa ke periode/*Sprint* selanjutnya.

3.2.4 Git Flow

```
Master
Develop
Hofixes/
Feature/
  BedahRumah/
    UserJourney/
      UJ-Sprint-6 ( parent ) ( mang yana )
      RA-236-Balance (Ticket-Title)
      RA-399-Panduan_Photo_KTP_and_Selfie
      RA-263-KYC_Status

      UJ-Sprint-7
      RA-211-QR

      UJ-Sprint-7-KYC // PIC needed ( Yana )
      UJ-Sprint-7-KYC_Status_Photo_KTP ( branch ) ( yana )
      [RA-255] KYC Status ( commit an )
      [RA-256] Photo KTP ( commit an )

      UJ-Sprint-7-KYC_Selfie_OCR ( branch ) ( Ari )
      [RA-260] Selfie ( commit an )
      [RA-279] OCR ( commit an )
    Backup/

    Transaction/
      TR-Sprint-6 ( parent ) ( sam Wisnu )
      TR-Sprint-6-Airtime ( branch ) ( rifky ) // Satu module
      [RA-90] InputID
      [RA-100] Denom
      [RA-200] Inquiry

      root branch - title
      TR-Sprint-6-Recurring
      TR-Sprint-6-Airtime_Denom
```

Gambar 3.16 Struktur Git Linkaja iOS
Linkaja Bedah Rumah Squad, Git Branch (2021)

Setiap Developer mempunyai ruang kerja masing-masing dalam mengerjakan modul dari sebuah tiket yang diambil sewaktu *Sprint Planning*, yang dimaksud ruang kerjanya menggunakan *Source Control Git*. Dalam mempunyai ruang kerja didalam *Branch*(cabang) *Git* terdapat *Guidelines* untuk memulai bekerja dalam ekosistem *Git Project* ini.

Sebagai contoh kasus saya di *Squad* Bedah Rumah, saya bekerja dalam ranah mengerjakan modul yang berkaitan semua dengan *UserJourney*, jadi saya memulai bekerja dengan membuat *Branch* baru dari basis *Parent Branch* yang terdapat didalam direktori *UserJourney*, yaitu *branch* UJ-Sprint-7.

Dan ketika saya sudah menyelesaikan *task* saya dalam mengerjakan suatu modul, saya akan melakukan *Merge Request* Kembali dengan target *Branch* yaitu basis dari *Branch* yang saya buat, UJ-Sprint-7.

Tentu saya memerlukan *Code Review* dari rekan-rekan saya yang bekerja didalam naungan yang sama di *UserJourney* untuk memindahkan

Changes Files dari *branch* saya ke *branch* yang tiada *Ownership* saya di *branch* ini, tentu harus memiliki persetujuan dalam melakukan perubahan *branch* yang bukan milik saya.

3.2.5 Code Review

Tentu dalam mengerjakan suatu modul terdapat proses dimana semua Developer memigrasi modul yang telah dikerjakan dari Branch ruang kerja individu masing-masing ke git *branch* asalnya, bisa disebut *Parent-Branchnya*. Dan dalam memigrasi modul dari *branch* kita, diperlukan *Approval*/persetujuan dari rekan kerja kita, tujuannya untuk mengontrol kualitas dari *Code* yang akan *dideliver* tiap individu ke *Production*. Dengan memberi masukan *Code* yang dapat diperbaiki ke lebih baik, contohnya suatu *code* rawan menjadi penyebab kebocoran memory, *Code* tidak efisien, *Code* tanpa sadar membuat duplikasi pembuatan Function/Object/Singleton yang fungsinya sama saja dengan yang sudah ada lalu diarahkan untuk menggunakan *Resources* yang sudah ada demi *Code* yang lebih baik.

The screenshot shows a GitLab Merge Request (MR) for the project 'LinkAja Consumer Apps'. The MR is titled 'Bug Fix Email Settings Defects' and was opened 2 weeks ago by Muhammad DaryA (Developer). It has 1 commit, 1 pipeline, and 4 changes. The MR is in a 'Merged' state. The description lists three issues: No 44 (warning on resend), No 60 (error message on keyboard input), and No 61 (button 'Kirim Ulang' not responding). The MR was approved by Ari Prasetyo and merged into the 'Feature/BedahRumah/UserJourney/Regress_Email' branch. The source branch has been deleted.

mobile > iOS > LinkAja Consumer Apps > Merge Requests > 1424

Merged Opened 2 weeks ago by Muhammad DaryA (Developer) Edit

— Bug Fix Email Settings Defects —

Overview 1 Commits 1 Pipelines 1 Changes 4

- No 44: Muncul warning saat resend valid email
- No 60: Error message ketika input email yg tidak sesuai/tidak terdaftar tertutup oleh keyboard user dan harus di minimize keyboardnya agar terlihat dan terbaca, hal ini membuat user jadi unaware dengan kesalahan yg dilakukan dan mengira fitur unusable
- No 61: button 'Kirim Ulang' untuk email verifikasi merespon dengan pesan "email tidak terdaftar", suspect alamat email yg dikirimkan masih dalam bentuk masking sehingga dianggap tidak valid

Request to merge Feature/BedahRumah/UserJourney/Regress_Email into Feature/BedahRumah/...

Pipeline #55255 skipped for 5ac63424 on Feature/BedahRumah/UserJourney/Regress_Email

Merge request approved. Approved by Ari Prasetyo

View eligible approvers

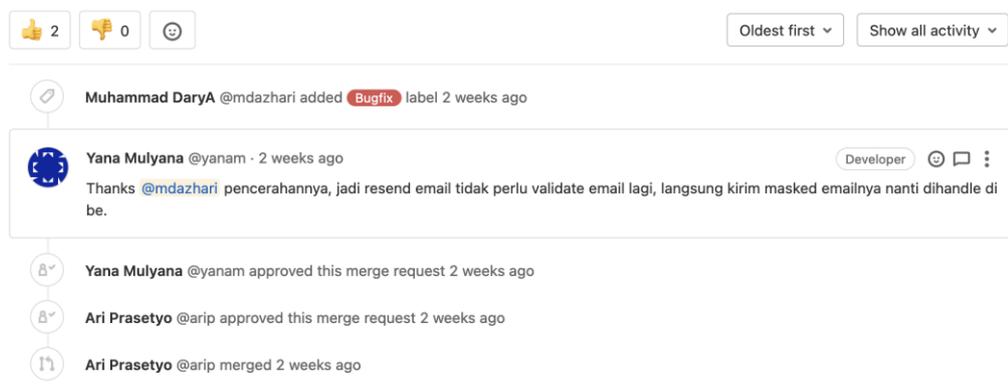
Merged by Ari Prasetyo 2 weeks ago

The changes were merged into Feature/BedahRumah/UserJourney/Regress with 5ac63424

The source branch has been deleted

Gambar 3.17 Proses *Merge Request* Tiket Modul Linkaja Bedah Rumah *Squad*, Gitlab Linkaja (2021)

Untuk melakukan *Code Review* dapat dilakukan dengan *Merge Request* dengan mengakses Gitlab Linkaja, memilih *Source branch* yakni ruang kerja kita dalam mengerjakan modul ke target *branch* yaitu *parent-branch* dari *source branch* yang akan melakukan permintaan *Merging/migrasi*.

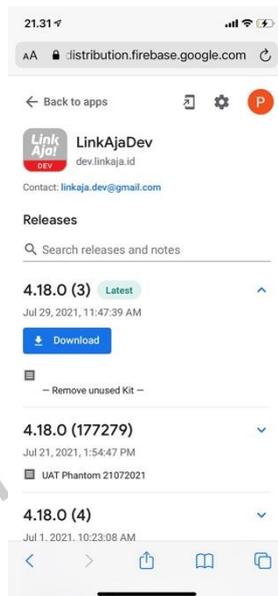


Gambar 3.18 Proses Code Review Tiket Modul
Linkaja Bedah Rumah Squad, Gitlab Linkaja (2021)

Approval dari *Code* yang sedang dalam fase *Code Review*, ditinjau minimal 2 orang. Setelah proses *Merge Request* sudah selesai dalam artian *Branch Merged*, maka sudah siap masuk ke fase pengetesan modul tersebut.

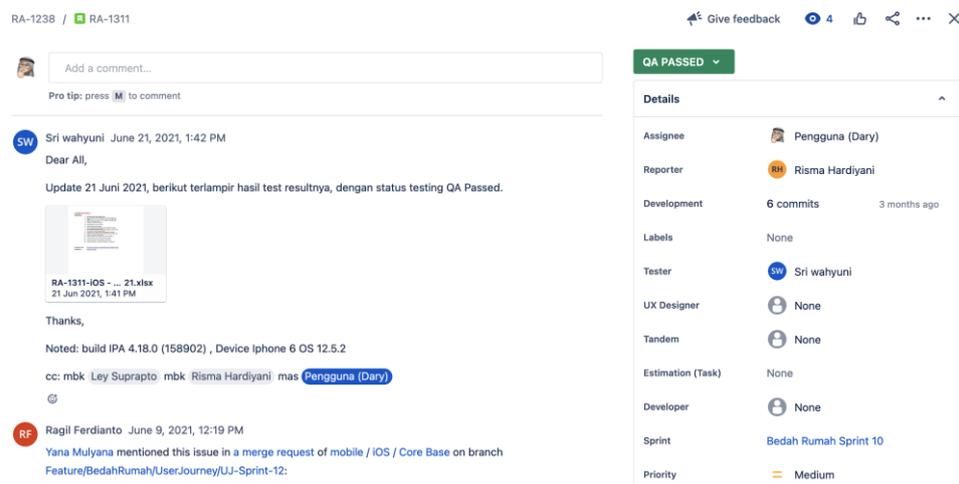
3.2.6 Pengetesan Tiap Modul

Setiap *Developer* yang memegang suatu tiket modul memiliki tanggung jawab untuk menyerahkan distribusi aplikasi kedalam *platform* *Firebase Distribution* untuk dites oleh *QA Tester* yang bersangkutan. Disini sudah tidak perlu *repot-repot* melakukan distribusi file aplikasi secara manual, tapi sudah adanya otomasi dimana setiap ketika suatu *Git Branch* melakukan *merging/migrasi* maka *branch* tersebut akan secara otomatis generate file aplikasi ke *Firebase Distribution*.



Gambar 3.19 Pendistribusian Modul Untuk Dites Oleh QA Linkaja Bedah Rumah Squad, Firebase Distribution (2021)

Istilah ini sering dikenal dengan sebutan CI/CD atau *Continuous Integration and Continuous Delivery/Continuous Deployment*.



Gambar 3.20 Proses Tiket Modul Dites Oleh QA Linkaja Bedah Rumah Squad, Draw.io (2021)

Tiket modul Jira yang sudah berhasil mendistribusikan aplikasi di Firebase Distribution bisa langsung diinformasikan ke QA Tester dengan cara memberi komentar *Build Notes* di tiket terkait lalu digeser status tiketnya menjadi *Ready For QA*. Dan ketika tiket tersebut yang sudah di *Self Assign* oleh QA Testernya sendiri akan menggeser statusnya menjadi

In QA sehingga sebagai Developer tiket tersebut dapat melakukan *follow-up* perkembangan dari pengetesan tiket tersebut.

Status	Counter	Severity	Counter	Action	Counter
New	0	High	#REF!	Required to fix prior Release	#REF!
Assigned	0	Medium	#REF!	Required to fix prior Regression Test	#REF!
Pending (On Hold)	0	Low	#REF!	Will be treated as Defect (Kanban)	#REF!
Fixing	0	Informational	#REF!	Others (please put notes)	#REF!
Ready for QA	0				
Testing	0				
Passed	2				
Passed with notes	0				
Failed	0				
Re-Test	0				
Not a Bug	0				
Rejected	0				
Duplicate	0				

Gambar 3.21 Dokumen Hasil Pengetesan Tiket Modul oleh QA Linkaja Bedah Rumah Squad, Excel (2021)

Di fase ini Developer dan QA Tester akan berkomunikasi mengenai persoalan pengetesan distribusi aplikasinya. Jika semua pengetesan berjalan sukses sesuai dengan *Acceptance Criteria*, maka status tiket akan digeser menjadi *QA Passed* dan QA yang terkait akan melampirkan hasil pengetesan tiket tersebut. Namun jika ada beberapa yang tidak sesuai dengan *Acceptance Criteria*, akan dikomunikasikan terlebih dahulu ke Developer jika itu *bug* yang valid, maka QA akan membuat beberapa *sub-task bug* dan tiket tersebut akan digeser statusnya menjadi *Bug*.

3.2.7 Pra Rilis



Gambar 3.22 Pendistribusian Aplikasi Final ke Internal Linkaja
Linkaja Bedah Rumah Squad, TestFlight (2021)

Setelah semua tiket berstatus *Done* dan *Sprint* sudah usai dan tidak ada tiket yang tersisa harus dikerjakan, maka akan menjadi sebuah tanggung jawab Tech Lead/Specialist untuk mendistribusikan sebuah aplikasi Linkaja yang mana modul-modul para Developer yang dikembangkan dikemas menjadi satu paket dan membawa sebuah distribusi dengan modul lengkap ke jenjang *Internal Testing* dari fase *Internal Alpha Testing* sampai *Internal Beta Testing* yang sering didengar bernama *Regression Testing*, yang bertujuan untuk memastikan bahwa sebuah modul lengkap pengembangan Aplikasi Linkaja yang akan di rilis ke pasar bebas *Bug*, dibantu dengan semua karyawan Linkaja untuk mengunduh *Internal Testing* Linkaja melalui aplikasi bernama TestFlight (untuk iOS) yang mana semua karyawan akan mendapatkan *Broadcast* email dan melapor jika terjadi sesuatu yang tidak normal.

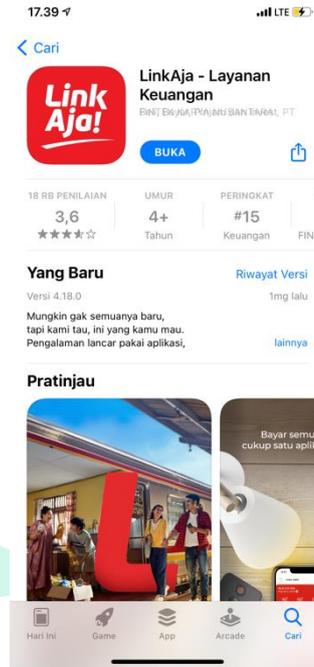
No	Journey / Submodule	Defects Description	Attachments - Link Defect Folder	Investigation Result	Ownership Defect	Developer
71	PGN	[EN] wording tidak sesuai masih 'ID pelanggan'	https://drive.google.com/file/d/1Nb2UUGNo-pq0wrbBimVhmM8nOHDGBZ5/view?usp=sharing	kasus terjadi sama dengan nomer 11	Ownership : Front End (Fixed)	Wianu
72	Telephone	[EN] wording pada page input misdn masih berbahasa indonesia 'Nomor Telepon' dan 'Telepon', wording pada Review & Confirm masih berbahasa Indonesia 'Tagihan'	https://drive.google.com/file/d/19H-m7agzkwDY2nPW6511MrCOIN0bdi/view?usp=sharing	Untuk input yang berbahasa Indonesia kasusnya sama dengan nomer 11 sedangkan wording Tagihan pada halaman Review & Confirm karena result dari BE.	Ownership : Front End & Back End	Wianu
73	Telephone	[ID] wording saat load masih berbahasa Inggris Loading..'	https://drive.google.com/file/d/1tcVtXaDkSk4W1OEKBykXoLl-XK2NzWLU0/view?usp=sharing		Ownership : Front End (Fixed)	Ritky
74	Insurance	[EN] Halaman asuransi, webview, notifikasi transaksi berhasil semua menggunakan bahasa indonesia	https://drive.google.com/drive/folders/1G2qZUPZKvPa01Xr9OrFu7gT3oCHc?usp=sharing	webview sudah ada parameter lang(language), untuk menu Bayar Asuransi akan ikut nomer 11	Ownership : Back End	Farizy
75	Telephone	[ID] status transaksi "success & failed" , wording loading masih berbahasa Inggris	https://drive.google.com/file/d/1Fnm_X_SryhNqAGzVayaQ-q74GxECAG/view?usp=sharing		Ownership : Front End (Fixed)	Yana
76	Pulsa/Data	[EN] Wording halaman gagal masih ada yang berbahasa Indonesia	https://drive.google.com/file/d/1GM85m9d5p-xLVYAP4Nr9w1gZSweleTZ/view?usp=sharing	Wording untuk deskripsi "Transaksi Telekomset Prepaid Gagal" dari BE, berikut url respon https://drive.google.com/file/d/1cUoIv7wianyar9hSPZzqg6Wii056Kq/view?usp=sharing	Ownership : Back End	Ritky

Gambar 3.23 Dokumen *Regression Test* Aplikasi Linkaja Final Linkaja Bedah Rumah Squad, Excel (2021)

Terdapat penemuan 76 *Defects List*/Daftar kerusakan modul beserta lampiran video bukti yang harus di investigasi oleh para Owner/Developer yang mengerjakan tiket terkait dan harus segera diperbaiki jika kerusakan itu hal yang valid. Jika sudah diperbaiki dan melakukan *Git Merging Branch* lalu Tech Lead akan mengemas perbaikan dari modul-modul tersebut dan pendistribusian ulang.

3.2.8 Rilis ke Pasar

Rilisnya aplikasi Linkaja dengan arsitektur yang baru ke pasaran dikedua platform Android & iOS tepat tanggal 28 Juli 2021 merupakan sebuah pencapaian yang luar biasa bagi kami semua.



Gambar 3.24 Tampilan Aplikasi Linkaja di App Store
Aplikasi Linkaja, App Store (2021)

Dengan perjuangan yang tentu tidak mudah, kontribusi bagi semua pihak membuahkan hasil yang sangat memuaskan.

3.3 Kendala Yang Dihadapi

Kendala yang dihadapi dalam mengerjakan Project ini yaitu saya diharuskan adaptasi yang cepat dan belajar dalam Develop modul-modul dari modul yang sudah ada di aplikasi sebelumnya maupun yang baru dengan Arsitektur baru ini, karena konsekuensi dari telatnya Project ini akan terputusnya perilisan pembaruan aplikasi (arsitektur lama) yang sudah rutin melakukan pembaruan sebulan sekali.

Karena saya sebelumnya berada di tim/Squad User Journey yang mana mengerjakan/mengembangkan fitur yang pengguna dapat lihat & rasakan dan semua orang developer Mobile yang di tim/Squad ini dipindahkan sehingga tidak akan ada pembaruan di aplikasi lama.

Dan proses belajar & adaptasi yang mana kewajiban saya harus menulis ulang Code ke arsitektur yang baru butuh waktu yang tidak sedikit untuk mempelajari dan beradaptasinya, walaupun hasilnya memang

sangat memuaskan, Project kini menjadi jauh lebih Scalable, Readable, Maintainable dibanding sebelumnya.

3.4 Cara Mengatasi Kendala

Cara mengatasi kendala ini tentu solusinya sebagai manusia sosial tentunya kita harus menjalin komunikasi kesesama tim ketika ada kendala atau masalah yang dihadapi, karena sesama tim pasti akan membantu dengan lewat jalur pribadi ataupun acara Sharing Session. Jadi diharapkan tidak akan ada yang melewati kesulitan sendirian.

