

**IMPLEMENTASI *FRAMEWORK* FLUTTER 3 UNTUK
MENGEMBANGKAN MODUL PEMBAYARAN DAN PELACAKAN
PENGIRIMAN PADA APLIKASI UNILAHUB MENGGUNAKAN
METODE *PROTOTYPING***

(Skripsi)

Oleh

**RIFAN SETIADI
NPM 2017051019**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**IMPLEMENTASI *FRAMEWORK* FLUTTER 3 UNTUK
MENGEMBANGKAN MODUL PEMBAYARAN DAN PELACAKAN
PENGIRIMAN PADA APLIKASI UNILAHUB MENGGUNAKAN
METODE *PROTOTYPING***

Oleh

RIFAN SETIADI

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung**



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2024**

DAFTAR ISI

	Halaman
DAFTAR ISI.....	i
DAFTAR TABEL	iii
DAFTAR GAMBAR.....	iv
I. PENDAHULUAN.....	5
1.1. Latar Belakang	5
1.2. Rumusan Masalah	7
1.3. Batasan Masalah.....	8
1.4. Tujuan.....	8
1.5. Manfaat Penelitian.....	8
II. TINJAUAN PUSTAKA	9
2.1. Penelitian Terdahulu.....	9
2.2. Uraian Tinjauan Pustaka	10
2.2.1. <i>E-Commerce</i>	10
2.2.2. Payment Gateway	11
2.2.3. Midtrans	11
2.2.4. RajaOngkir API	12
2.2.5. Flutter 3.....	12
2.2.6. Dart	13
2.2.7. Library GetX.....	14
2.2.8. GetX Pattern	14

2.2.9. Metode Pengembangan <i>Prototyping</i>	15
III. METODOLOGI PENELITIAN	18
3.1. Waktu dan Tempat Penelitian	18
3.2. Alat Penelitian	18
3.2.1. Perangkat Lunak (<i>Software</i>)	18
3.2.2. Perangkat Keras (<i>Hardware</i>)	19
3.3. Metode Pengumpulan Data	19
3.3.1. Data Primer	19
3.3.2. Data Sekunder	19
3.4. Tahapan Penelitian	20
3.4.1. Identifikasi Masalah.....	20
3.4.2. Studi Literatur	20
3.4.3. Pengembangan Aplikasi Menggunakan Metode <i>Prototyping</i>	21
3.4.4. Pembuatan Laporan	34
DAFTAR PUSTAKA	35

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu	9
2. Kebutuhan Pengguna.	21

DAFTAR GAMBAR

Gambar	Halaman
1. Struktur Direktori GetX Pattern.	15
2. SDLC Prototyping (Nursaid dkk., 2020).	16
3. Tahapan Penelitian.	20
4. Use Case Diagram UnilaHub Research Apps.	23
5. Use Case Diagram UnilaHub Public Apps.	23
6. Rancangan Struktur Direktori Kode Program.	25
7. Logical Data Model Fitur Pembayaran.	26
8. Rancangan Tampilan List Metode Pembayaran.	27
9. Rancangan Tampilan Detail Metode Pembayaran.	28
10. Rancangan Tampilan Halaman Pembayaran Berhasil.	29
11. Rancangan Tampilan Pelacakan Pengiriman.	30
12. Tabel categories yang Telah Ada.	31
13. Rancangan Perubahan Struktur Tabel categories.	31
14. Logical Data Model Fitur Pelacakan Pengiriman.	32

I. PENDAHULUAN

1.1. Latar Belakang

Aplikasi UnilaHub merupakan aplikasi *e-commerce* berbasis *mobile* milik Universitas Lampung yang dikembangkan pada tahun 2022. Aplikasi UnilaHub sendiri tersedia dalam dua *platform* yang berbeda yaitu “UnilaHub (*Public Apps*)” dan “UnilaHub (*Research Apps*)”. *Platform* aplikasi “UnilaHub (*Public Apps*)” adalah aplikasi yang dikembangkan sebagai bentuk upaya mempermudah Universitas Lampung dalam mempublikasikan dan memasarkan produk hasil penelitian yang ada (Alawi, 2022). *Platform* “UnilaHub Research Apps” menjadi wadah bagi peneliti di Universitas Lampung untuk melakukan hilirisasi produk-produk hasil penelitian kepada masyarakat dan dunia industri (Permana, 2022).

Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Lampung bertanggung jawab meningkatkan kualitas penelitian dan kinerja pengabdian kepada masyarakat dengan menyediakan berbagai skema penelitian dan pengabdian khusus bagi dosen Universitas Lampung. Tanggung jawab LPPM serta Tingkat Kesiaapterapan Teknologi (TKT) yang ditetapkan oleh Direktorat Riset dan Pengabdian kepada Masyarakat (DPRM) menjadi dasar pengembangan aplikasi UnilaHub sebagai tolak ukur sistematis bagi produk hasil penelitian sehingga dapat diadopsi oleh pengguna, baik oleh pemerintah, industri, atau masyarakat. Selain itu, hasil riset penelitian yang telah sampai pada tahap hilirisasi akan dapat meningkatkan Indikator Kinerja Utama perguruan tinggi.

Proses pembayaran produk atau jasa pada *platform* “UnilaHub (*Public Apps*)” saat ini masih dilakukan diluar aplikasi itu sendiri, yaitu dengan cara mengirim uang melalui ATM (Anjungan Tunai Mandiri) atau melalui *mobile banking*. Setelah melakukan pembayaran, pembeli harus mengunggah bukti pembayaran ke dalam

aplikasi yang kemudian akan di konfirmasi oleh admin. Hal tersebut mengurangi efisiensi kegiatan jual beli pada aplikasi yang mana pembayaran tidak dapat dikonfirmasi secara langsung oleh sistem. Selain itu, proses pelacakan pengiriman perlu diimplementasikan pada aplikasi “UnilaHub (*Public Apps*)” maupun “UnilaHub (*Research Apps*)” untuk memudahkan pembeli serta penjual melihat status dan keberadaan produk yang dikirimkan.

Payment Gateway adalah suatu sistem pembayaran elektronik yang memungkinkan pelanggan melakukan transaksi pembayaran secara *online* melalui aplikasi atau *website* serta memfasilitasi pengecekan pembayaran secara *real-time* (Ningsih dan Sanjaya, 2023). Implementasi *Payment Gateway* pada *platform* “UnilaHub (*Public Apps*)” dapat menjadi solusi alternatif pembayaran sehingga pelanggan tidak perlu menunggu konfirmasi pembayaran oleh admin. Integrasi *Payment Gateway* merupakan langkah penting untuk meningkatkan keamanan dan efisiensi transaksi.

Selain fitur pembayaran, salah satu hal yang menjadi sorotan pada aplikasi UnilaHub baik *platform Public Apps* maupun *Research Apps* adalah pelacakan pengiriman. Ketidakpastian dalam waktu pengiriman sesuai dengan jangka waktu yang dijanjikan oleh layanan, bersama dengan kurangnya informasi tentang lokasi terkini barang kiriman, membuat konsumen kesulitan memahami status layanan pengiriman yang sudah mereka bayar (Eviani dan Hidayat, 2021). Ketepatan informasi perjalanan produk yang sedang dikirimkan terhadap ekspektasi pelanggan menjadi faktor kepuasan tersendiri bagi pelanggan. Implementasi pelacakan pengiriman dengan menggunakan pihak ketiga pada aplikasi UnilaHub menjadi solusi alternatif yang diharapkan dapat meningkatkan kepuasan pengguna terhadap layanan yang diberikan.

Integrasi pengembangan aplikasi UnilaHub dengan metode pengembangan tertentu menjadi hal yang perlu diperhatikan. Penggunaan metode *prototyping* dalam pengembangan perangkat lunak memiliki keunggulan dalam memperoleh umpan balik pengguna secara cepat terhadap versi awal aplikasi (*prototype*) yang dikembangkan (Nugraha dan Syarif, 2018). Tujuan dari penggunaan metode *prototyping* adalah untuk menyempurnakan aplikasi *prototype* berikutnya berdasarkan pada evaluasi kebutuhan pengguna.

Salah satu *framework* yang menjadi fokus perhatian dalam pengembangan aplikasi *mobile* adalah Flutter. Flutter adalah *framework open source* dari Google yang memberikan kemudahan bagi pengembang untuk membangun sebuah aplikasi yang dapat berjalan pada berbagai *platform* seperti Android, IOS, Desktop dan yang lainnya tanpa harus membuat kode program yang berbeda (Flutter, 2023). Flutter versi 3 membawa sejumlah pembaruan dan peningkatan yang signifikan, menciptakan peluang baru bagi pengembang dalam meningkatkan fungsionalitas dan performa aplikasi.

Fokus penelitian ini adalah pada pengembangan modul pembayaran menggunakan *Payment Gateway* pada *platform* “UnilaHub (*Public Apps*)” serta modul pelacakan pengiriman untuk kedua *platform* UnilaHub menggunakan pihak ketiga, yang akan diimplementasikan dengan memanfaatkan kelebihan Flutter versi 3 dan metode pengembangan *prototyping*. Penelitian ini melanjutkan pengembangan sebelumnya pada aplikasi UnilaHub yang masih menggunakan Flutter versi 2, dengan menambahkan fitur-fitur yang menjadi kebutuhan utama dalam pengalaman berbelanja online.

Berdasarkan uraian latar belakang yang ada maka ditetapkan penelitian di Universitas Lampung dengan judul **“Implementasi *Framework* Flutter 3 Untuk Mengembangkan Modul Pembayaran Dan Pelacakan Pengiriman Pada Aplikasi Unilahub Menggunakan Metode *Prototyping*”**.

1.2. Rumusan Masalah

Berdasarkan pemaparan latar belakang yang ada maka masalah yang dirumuskan pada penelitian ini adalah tentang bagaimana mengembangkan modul pembayaran menggunakan *Payment Gateway* dan pelacakan pengiriman pada aplikasi UnilaHub dengan mengimplementasikan metode pengembangan *prototyping* dengan basis *framework* Flutter versi 3.

1.3. Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Penelitian ini berfokus pada pengembangan modul pembayaran untuk *platform* “UnilaHub (*Public Apps*)” dan pelacakan pengiriman untuk *platform* “UnilaHub (*Public Apps*)” maupun “UnilaHub (*Research Apps*)”.
2. Pada penelitian ini migrasi kode program dari Flutter versi 2 ke dalam Flutter versi 3 dilakukan untuk *platform* “UnilaHub (*Research Apps*)”.
3. *Framework* yang digunakan adalah Flutter versi 3.
4. API (*Application Programming Interface*) yang digunakan merupakan API yang telah dikembangkan pada penelitian sebelumnya.
5. Penambahan maupun pengubahan API (*Application Programming Interface*) dilakukan pada bagian yang berhubungan dengan penambahan modul pembayaran dan pelacakan pengiriman.

1.4. Tujuan

Tujuan penelitian ini adalah mengembangkan modul pembayaran pada *platform* “UnilaHub (*Public Apps*)” dan pelacakan pengiriman pada kedua *platform* aplikasi UnilaHub menggunakan *framework* Flutter versi 3.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut.

1. Memberikan kemudahan bagi pembeli untuk melakukan transaksi pembayaran secara langsung di dalam aplikasi UnilaHub.
2. Membantu pembeli dan peneliti yang dalam konteks ini adalah penjual untuk memantau dan melacak pengiriman produk.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu berperan sebagai pembanding antara penelitian yang telah dilakukan dengan penelitian yang akan dilakukan. Selain itu, penelitian terdahulu juga dapat dijadikan sebagai rujukan untuk mendukung penelitian yang akan dilakukan. Berikut adalah beberapa penelitian terdahulu yang dijadikan pembanding sekaligus rujukan dalam penelitian ini.

Tabel 1. Penelitian Terdahulu

No.	Judul Penelitian	Metode	Hasil
1	Pengembangan Aplikasi <i>Mobile E Commerce</i> Produk Riset Universitas Lampung Menggunakan Metode Scrum (<i>Public Apps</i>)	Metode Scrum	Aplikasi <i>Marketplace</i> Berbasis Android (UnilaHub Public Apps)
2	Hilirisasi Produk Penelitian Universitas Lampung Menggunakan Platform <i>E-Commerce</i> Berbasis <i>Mobile</i> (<i>Research Apps</i>)	Metode Scrum	Aplikasi <i>Marketplace</i> Berbasis Android (UnilaHub Researcher)

Tabel 1. (Lanjutan)

No.	Judul Penelitian	Metode	Hasil
3	Implementasi <i>Payment Gateway</i> Menggunakan Midtrans Pada <i>Marketplace</i> Travnesia.com	<i>Unified Modeling Language</i> (UML)	Aplikasi <i>Marketplace</i> Travnesia.com Yang Terintegrasi Dengan <i>Payment Gateway</i> Midtrans
4	<i>E-Commerce Marketplace</i> Petshop Menggunakan Integrasi RajaOngkir API dan iPaymu <i>Payment Gateway</i> API	Metode SDLC Dengan Pendekatan <i>Waterfall</i>	Sistem Informasi Berbasis <i>Website</i> yang terintegrasi dengan RajaOngkir API dan iPaymu <i>Payment Gateway</i> API
5	<i>E-Commerce</i> Pakaian Berbasis Android Menggunakan Metode <i>Prototype</i>	Metode <i>Prototyping</i>	Aplikasi <i>E-Commerce</i> Pakaian Berbasis Android

2.2. Uraian Tinjauan Pustaka

2.2.1. *E-Commerce*

E-commerce merujuk pada aktivitas penyebaran, pembelian, penjualan, serta pemasaran produk dan layanan oleh individu melalui *platform* elektronik seperti internet, WWW (*World Wide Web*), atau jaringan komputer lainnya. *E-commerce* dapat mencakup transfer data elektronik, pertukaran informasi elektronik, pengelolaan inventori otomatis, dan proses pengumpulan data secara otomatis (Listianto dkk., 2017). *E-commerce* merupakan bentuk pemasaran produk baik barang maupun jasa melalui sistem informasi. Selain itu, *e-commerce* juga memungkinkan pengguna untuk melakukan pembayaran dan pelacakan pengiriman barang melalui sistem informasi.

2.2.2. Payment Gateway

Menurut Ningsih dan Sanjaya (2023), *payment gateway* merupakan sistem yang diadopsi oleh layanan *e-commerce* untuk memberikan kemudahan dalam melakukan proses pembayaran secara *real-time*. Tujuan utama dari sistem *payment gateway* adalah untuk memfasilitasi verifikasi transaksi secara cepat dan efisien. Melalui *payment gateway*, penjual dapat secara langsung melakukan pengecekan pembayaran, memberikan kepastian transaksi, dan mengoptimalkan kepuasan berbelanja *online* bagi konsumen. Hal tersebut memberikan keleluasaan bagi pelanggan maupun penjual dalam memperluas cakupan transaksi tanpa adanya kendala waktu dan tempat. Selain itu, dengan menggunakan *payment gateway* keamanan transaksi lebih terjamin dimana verifikasi pembayaran akan dikonfirmasi secara otomatis oleh sistem.

Pemerintah Indonesia sendiri telah mendukung penggunaan sistem *payment gateway* sebagai media atau alat pembayaran pada layanan *e-commerce*. Beberapa Fintech Payment Gateway yang didukung di Indonesia antara lain adalah Midtrans, Doku dan Xendit.

2.2.3. Midtrans

Midtrans merupakan perusahaan teknologi keuangan di Indonesia yang berdiri pada tahun 2012. Midtrans berfokus pada sektor finansial dengan menyediakan solusi *Payment Gateway*. Perusahaan ini menawarkan dukungan untuk beragam metode pembayaran *online* dan layanan kirim dana (*disbursement*) untuk mempermudah pelanggan dalam melakukan transaksi bisnis (Midtrans, 2023).

Midtrans mengimplementasikan sejumlah langkah-langkah keamanan untuk melindungi data transaksi pengguna. Pertama, Midtrans telah terdaftar dan memegang lisensi Bank Indonesia sejak 2018. Kedua, Midtrans telah terdaftar di Kementerian Komunikasi dan Informatika Republik Indonesia (Kominfo) sebagai penyelenggara Sistem Elektronik. Ketiga, Midtrans mengimplementasikan enkripsi AES-256 milik U.S. National Institute of Standards and Technology yang

merupakan standar enkripsi elektronik untuk melindungi data sensitif dari sistem yang tidak diotorisasi. Keempat, Midtrans memiliki lisensi Payment Card Industry Data Security Standard (PCI-DSS) yang menunjukkan bahwa sistem Midtrans telah memenuhi standar keamanan data yang tinggi. Kelima, Midtrans telah bersertifikasi ISO/IEC 27001.

2.2.4. RajaOngkir API

RajaOngkir adalah *platform* dan layanan *web* (API) yang menyediakan informasi biaya pengiriman dari berbagai perusahaan pengiriman di Indonesia seperti POS Indonesia, JNE, TIKI, PCP, ESL, dan RPX (RajaOngkir, 2023). Secara umum, RajaOngkir dirancang untuk memberikan informasi dan perbandingan biaya pengiriman dari berbagai perusahaan pengiriman, khususnya bagi pemilik toko online dan individu yang sering berbelanja *online*.

Keunikan sistem RajaOngkir terletak pada integrasi data. Pengguna hanya perlu memasukkan sekali informasi seperti nama kota pengirim, kota penerima, dan berat paket, RajaOngkir secara otomatis memeriksa tarif dari semua perusahaan pengiriman yang didukung sehingga membantu menghemat waktu dan tenaga pengguna. RajaOngkir juga menyediakan antarmuka pemrograman aplikasi (API) biaya pengiriman yang dapat dimanfaatkan oleh pengembang untuk membuat aplikasi di berbagai platform, termasuk Android, BlackBerry, iOS, desktop, dan lainnya. Untuk menjaga akurasi data, data ongkos kirim yang diberikan oleh RajaOngkir diperoleh langsung dari *web* seluruh perusahaan ekspedisi pengiriman (Putra dkk., 2020).

2.2.5. Flutter 3

Flutter adalah *framework open source* dari Google yang memberikan kemudahan bagi pengembang untuk membangun sebuah aplikasi yang dapat berjalan pada berbagai platform seperti Android, IOS, Desktop dan yang lainnya tanpa harus membuat kode program yang berbeda (Flutter, 2023).

Flutter versi 3 membawa sejumlah pembaruan dan peningkatan yang signifikan, menciptakan peluang baru bagi pengembang dalam meningkatkan fungsionalitas dan performa aplikasi. Pada Flutter versi 3, penggunaan Flutter untuk mengembangkan sistem berbasis macOS, Linux dan Windows telah stabil (Flutter, 2022). Menurut situs dokumentasi resmi milik Flutter (Flutter Documentation), berikut adalah beberapa perubahan yang diimplementasikan pada Flutter versi 3.

1. Input teks internasional, termasuk untuk bahasa yang menggunakan editor metode input teks (IME) seperti Bahasa Tionghoa, Jepang, dan Korea, sepenuhnya didukung pada ketiga *platform*, termasuk metode input pihak ketiga seperti Sogou dan Google Japanese Input.
2. Layanan aksesibilitas yang mendukung layanan aksesibilitas seperti pembaca layar, navigasi yang dapat diakses, dan warna terbalik pada Windows, macOS, dan Linux.
3. Sejak Flutter 3, aplikasi desktop Flutter untuk macOS dibangun sebagai binary universal, dengan dukungan *native* untuk Mac berbasis Intel yang sudah ada dan perangkat Apple Silicon terbaru dari Apple.
4. Flutter 2.0 memperkenalkan *null safety*, fitur baru yang membantu pengembang menghindari kesalahan referensi *null*. *Null safety* adalah fitur yang kuat yang memungkinkan pengembang menulis kode yang lebih aman, lebih dapat diandalkan, dan telah diperbaiki lebih lanjut dalam Flutter 3.0. *Null safety* memungkinkan pengembang untuk dapat menulis kode yang lebih efisien, lebih mudah dibaca, dan kurang rentan terhadap kesalahan.
5. Sesuai dengan Kebijakan Deprecation Flutter, API yang sudah tidak lagi digunakan setelah rilis stabil 2.10 telah dihapus.

2.2.6. Dart

Dart adalah bahasa pemrograman berorientasi objek (*Object Oriented Programming*) yang mengadopsi gaya penulisan *C-Style* (Syaputra dan Yusmi, 2019). Dart memungkinkan pengembang untuk membuat antarmuka yang indah dan berkualitas pada device yang berbeda hanya dengan menggunakan satu bahasa pemrograman. Dart menjadi bahasa pemrograman yang diadopsi oleh Flutter.

2.2.7. Library GetX

GetX adalah *library* yang ringan dan powerful untuk Flutter (Pub.dev, 2023). GetX menyediakan berbagai fungsionalitas dan alat bantu untuk pengembangan aplikasi seperti *state management*, *navigation management* dan *dependencies management*. GetX memiliki 3 prinsip dasar, antara lain sebagai berikut.

1. Performa

GetX berfokus pada peningkatan performa dan konsumsi *resource* yang minimum serta tidak menggunakan Stream atau ChangeNotifier.

2. Produktifitas

Pada konteks manajemen memori, GetX menyederhanakan penghapusan *controller*, secara otomatis menghapus *resource* yang tidak digunakan, mengurangi kebutuhan untuk memikirkan pengelolaan memori secara eksplisit. Dependensi yang ingin disimpan dalam memori dapat dideklarasikan secara eksplisit dengan menambahkan opsi "permanent: true". Pendekatan ini tidak hanya efisien dalam penggunaan memori, tetapi juga mengurangi risiko menyimpan dependensi yang tidak diperlukan.

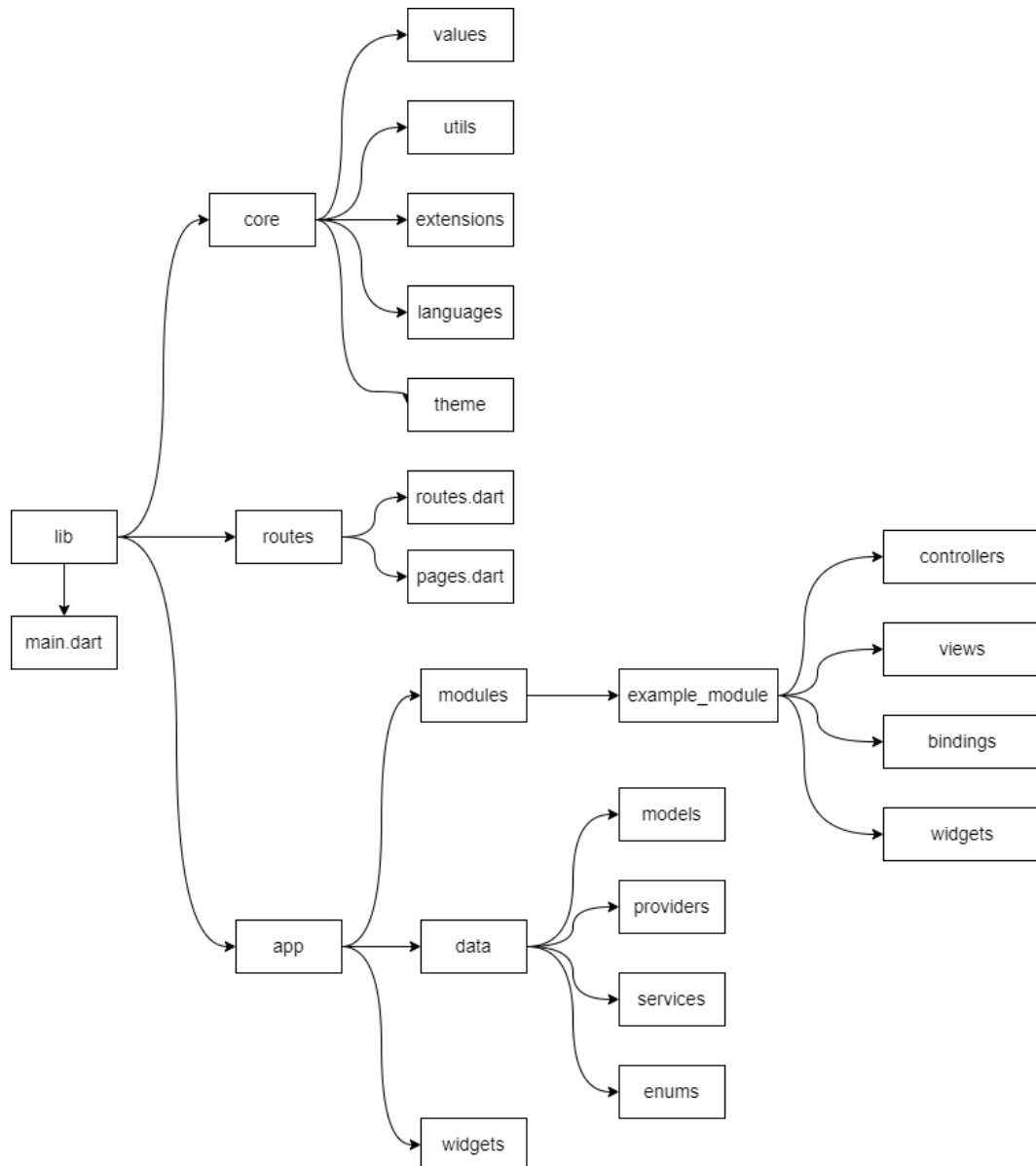
3. Organisasi

GetX mempermudah pemisahan *View*, *Presentation Logic*, *Business Logic*, *Dependency Injection*, dan Navigasi tanpa memerlukan konteks untuk berpindah antar halaman. Tidak seperti metode tradisional menggunakan widget tree, GetX memisahkan *presentation logic* dan *business logic* dari lapisan visual secara total. Fitur *dependency injection* yang dimiliki oleh GetX memungkinkan penggunaan untuk tidak perlu menyertakan kelas *Controller/Model/Bloc* ke dalam *widget tree* melalui *multiprovider*.

2.2.8. GetX Pattern

GetX Pattern adalah pola pengembangan (*design pattern*) yang dikembangkan untuk menstandarisasi proyek Flutter yang mengadopsi *library* GetX (GetX Pattern, 2023). Tujuan utama dari GetX Pattern adalah untuk membuat proyek Flutter dan GetX mengacu pada standar pengembangan yang harus diikuti sehingga

memudahkan dalam pengembangan selanjutnya. Berikut adalah struktur direktori dan berkas pada GetX Pattern.



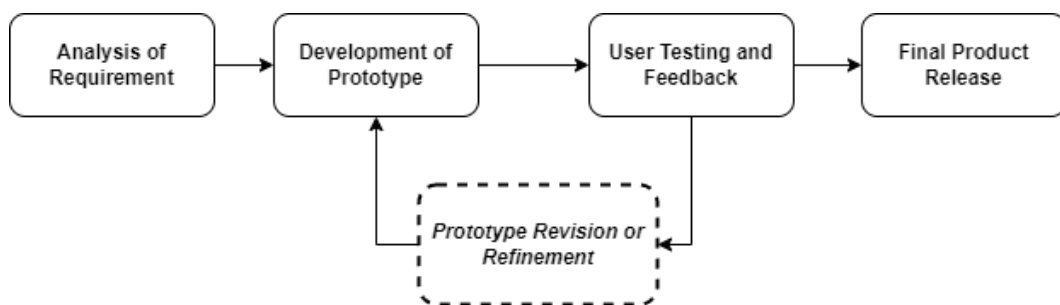
Gambar 1. Struktur Direktori GetX Pattern.

2.2.9. Metode Pengembangan *Prototyping*

Prototyping merupakan salah satu metode Siklus Hidup Pengembangan Sistem (System Development Life Cycle/SDLC), di mana proses pengembangan perangkat pada model ini mengalami iterasi dan dianggap lebih gesit untuk

pengembangan perangkat lunak (Nursaid dkk., 2020). Pendekatan *prototyping* menitikberatkan pada pemahaman kebutuhan pengguna, karena umpan balik dari pengguna dianggap sangat krusial dalam pengembangan perangkat lunak. Iterasi yang terjadi di dalam model atau metode *prototyping* merupakan bentuk penyempurnaan perangkat lunak berdasarkan pada umpan balik yang diberikan oleh pengguna.

Penggunaan metode *prototyping* dalam pengembangan perangkat lunak memiliki keunggulan dalam memperoleh umpan balik pengguna secara cepat terhadap versi awal aplikasi (*prototype*) yang dikembangkan (Nugraha dan Syarif, 2018). Tujuan dari penggunaan metode *prototyping* adalah untuk menyempurnakan aplikasi *prototype* berikutnya berdasarkan pada evaluasi kebutuhan pengguna.



Gambar 2. SDLC *Prototyping* (Nursaid dkk., 2020).

Tahapan yang dilakukan pada metode *prototyping* adalah sebagai berikut.

1. Analisis Kebutuhan (*Analysis of Requirements*)

Tahap pertama dalam metode *prototyping* adalah analisis kebutuhan. Tim pengembangan pada tahap ini akan bekerja sama dengan pemangku kepentingan untuk mengidentifikasi dan memahami kebutuhan dasar sistem yang akan dikembangkan. Diskusi mendalam dilakukan untuk menentukan fitur dan fungsi utama yang harus dimasukkan dalam prototipe.

2. Pengembangan Prototipe (*Development of Prototype*)

Setelah kebutuhan sistem dikumpulkan, tim mulai membangun prototipe awal. Prototipe ini dapat berupa model visual atau model perangkat lunak

yang mencakup sebagian fungsi sistem. Fokus pada tahap ini adalah untuk memberikan representasi yang cukup baik dari solusi yang diinginkan.

3. Uji Pengguna dan Umpan Balik (*User Testing and Feedback*)

Prototipe yang telah dibangun kemudian diuji oleh pengguna dan pemangku kepentingan terkait. Selama uji coba, mereka memberikan umpan balik terhadap desain dan fungsionalitas prototipe. Tujuan dari tahap ini adalah untuk mendapatkan perspektif langsung dari pengguna terkait dengan apakah prototipe ini memenuhi kebutuhan dan ekspektasi mereka.

4. Revisi atau Pemurnian Prototipe (*Prototype Revision or Refinement*)

Berdasarkan umpan balik yang diterima dari pengujian pengguna, tim pengembangan melakukan revisi atau pemurnian terhadap prototipe. Hal ini dapat melibatkan penambahan, penghapusan, atau perubahan fitur untuk meningkatkan kualitas dan kelayakan sistem. Proses ini dapat diulang beberapa kali hingga prototipe memenuhi persyaratan dengan baik.

5. Rilis Produk Akhir (*Final Product Release*)

Setelah prototipe telah melalui serangkaian revisi dan pemurnian, versi akhir dari produk dikembangkan. Produk akhir ini mencakup semua fitur dan fungsionalitas yang telah disepakati. Produk tersebut kemudian dirilis secara resmi ke pasar atau digunakan secara internal sesuai dengan kebutuhan organisasi.

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang beralamat di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1, Kota Bandar Lampung. Waktu pelaksanaan penelitian dimulai sejak bulan Desember 2023 hingga Maret 2024.

3.2. Alat Penelitian

Pada penelitian ini digunakan dua jenis alat yaitu perangkat lunak dan perangkat keras. Berikut merupakan rincian alat yang digunakan selama penelitian.

3.2.1. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan selama penelitian adalah sebagai berikut.

- a. Sistem Operasi Windows 11 Home Single Language 64-bit
- b. Android Studio versi Giraffe 2022.3.1
- c. Flutter versi 3.13.9
- d. Dart versi 3.1.5
- e. Postman versi 10.20.7
- f. draw.io
- g. Figma

3.2.2. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan selama penelitian adalah sebagai berikut.

- a. Laptop dengan spesifikasi:
 - 1) *System Manufacturer*: Acer
 - 2) *System Model*: Nitro AN515-44
 - 3) *Processor*: AMD Ryzen 5 4600H with Radeon Graphics (12 CPUs), 3.0 GHz
 - 4) *Memory*: 16,00 GB
- b. *Smartphone* dengan spesifikasi:
 - 1) *Device Model*: Realme C55
 - 2) *Processor*: MediaTek Helio G88
 - 3) RAM: 8,00 GB
 - 4) Sistem Operasi: Android 13

3.3. Metode Pengumpulan Data

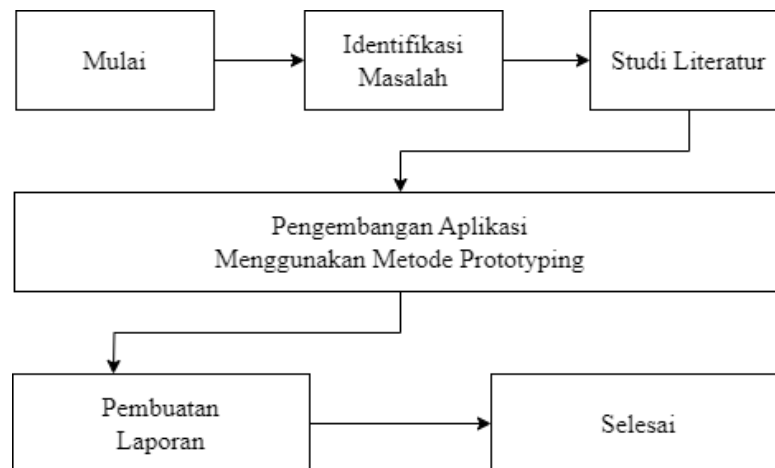
3.3.1. Data Primer

Data primer diperoleh dari hasil diskusi dan wawancara langsung dengan pihak Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Lampung.

3.3.2. Data Sekunder

Data sekunder didapatkan dari studi literatur seperti buku, jurnal, artikel, dan situs resmi. Situs resmi tersebut menyajikan informasi tentang implementasi *framework* Flutter, *library* GetX, GetX Pattern, Github, RajaOngkir API serta *Payment Gateway* Midtrans.

3.4. Tahapan Penelitian



Gambar 3. Tahapan Penelitian.

Penelitian ini dilakukan dalam empat tahap, yaitu Identifikasi Masalah, Studi Literatur, Pengembangan Aplikasi Menggunakan Metode *Prototyping*, dan Pembuatan Laporan.

3.4.1. Identifikasi Masalah

Penelitian ini dimulai dengan tahap identifikasi masalah yang menjadi dasar utama dalam pengembangan sistem menggunakan Metode *Prototyping*. Identifikasi masalah dilakukan oleh tim pengembang untuk memahami permasalahan yang ada dan mengumpulkan informasi yang relevan.

3.4.2. Studi Literatur

Sebelum memulai fase pengembangan perangkat lunak, tahapan studi literatur merupakan langkah awal yang krusial. Studi literatur dilakukan untuk menggali penelitian-penelitian terdahulu yang berkaitan dengan permasalahan yang sedang dihadapi. Tujuan dilakukannya studi literatur adalah agar penelitian ini dapat menjadi penelitian yang lebih baik.

3.4.3. Pengembangan Aplikasi Menggunakan Metode *Prototyping*

Metode yang digunakan dalam pengembangan aplikasi UnilaHub baik pada *platform Public Apps* maupun *Research Apps* adalah metode *prtotyping*. Tahapan yang terdapat pada metode *prototyping* adalah Analisis Kebutuhan, Pengembangan Prototipe, Pengujian serta Umpan Balik Pengguna, Perbaikan Prototipe, dan Perilisan Produk Akhir.

Tahap pengembangan dimulai dengan melakukan analisis kebutuhan, kemudian akan dilakukan pengembangan aplikasi dalam sebuah iterasi. Pada iterasi tersebut akan dimulai dengan pengembangan prototipe aplikasi yang kemudian akan diuji langsung oleh pengguna. Apabila prototipe belum sesuai dengan yang dibutuhkan pengguna maka akan dilakukan perbaikan terhadap prototipe tersebut. Iterasi tersebut dilakukan hingga prototipe yang dikembangkan sudah sesuai kebutuhan pengguna. Hasil akhir dari sebuah iterasi adalah *Final Product* yaitu aplikasi yang telah sesuai dengan kebutuhan pengguna.

3.4.3.1. Kebutuhan Pengguna

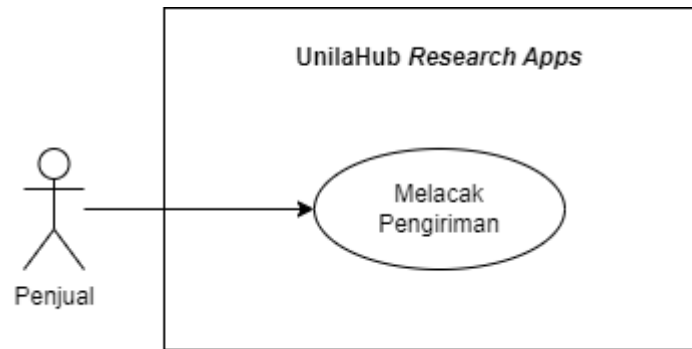
Berdasarkan pada hasil wawancara dengan pengguna aplikasi serta observasi secara langsung terhadap aplikasi UnilaHub yang telah dikembangkan sebelumnya, didapatkan kebutuhan yang dipaparkan dalam Tabel 2 berikut.

Tabel 2. Kebutuhan Pengguna.

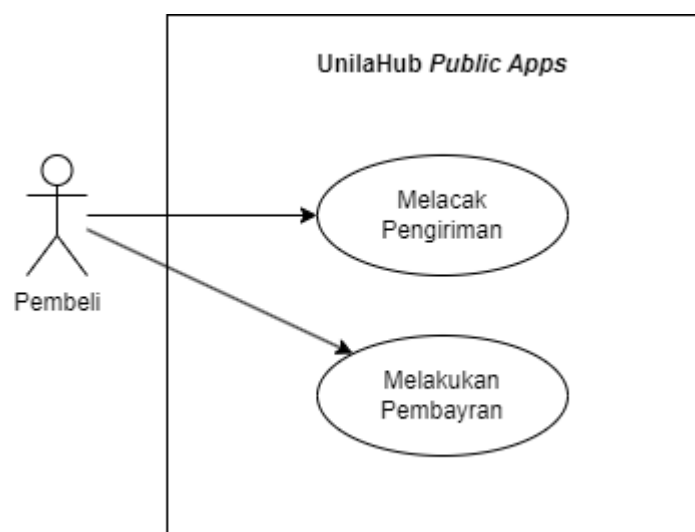
No.	Nama	Deskripsi
1	Migrasi Kode Program	Migrasi kode program yang sebelumnya ditulis menggunakan <i>framework</i> Flutter versi 2 menjadi Flutter versi 3.

No.	Nama	Deskripsi
2	<i>Redesign</i> Tampilan	Mengubah beberapa tampilan aplikasi agar menjadi lebih baru dan <i>fresh</i> dimata pengguna sesuai dengan penggunaan teknologi terbaru dari Flutter yaitu Material Design 3.
3	Penambahan Fitur atau Modul Pembayaran Pada Platform “UnilaHub (<i>Research Apps</i>)”	Fitur pembayaran merupakan fitur yang memungkinkan pengguna untuk melakukan pembayaran tanpa memerlukan verifikasi manual dari admin. Status pembayaran akan diverifikasi langsung oleh sistem atau aplikasi.
4	Penambahan Fitur Pelacakan Pengiriman Pada Kedua Platform Aplikasi	Fitur pelacakan pengiriman merupakan fitur yang memberikan informasi kepada pengguna terkait status pengiriman produk yang berbentuk barang secara <i>real-time</i> . Pengguna dapat mengetahui di mana paket berada dan kapan diperkirakan akan tiba.

Berdasarkan kebutuhan pengguna yang telah dipaparkan dalam Tabel 2, maka dihasilkan *Use Case Diagram* yang ditampilkan dalam Gambar 4 dan Gambar 5 berikut.



Gambar 4. *Use Case Diagram UnilaHub Research Apps.*



Gambar 5. *Use Case Diagram UnilaHub Public Apps.*

3.4.3.2. Rancangan Pengembangan Aplikasi

Berdasarkan pada kebutuhan pengguna dan *Use Case Diagram* yang telah dipaparkan, berikut adalah beberapa rancangan pengembangan yang akan dilakukan dalam penelitian ini.

a. Rancangan Migrasi Kode Program Platform UnilaHub (*Research Apps*)

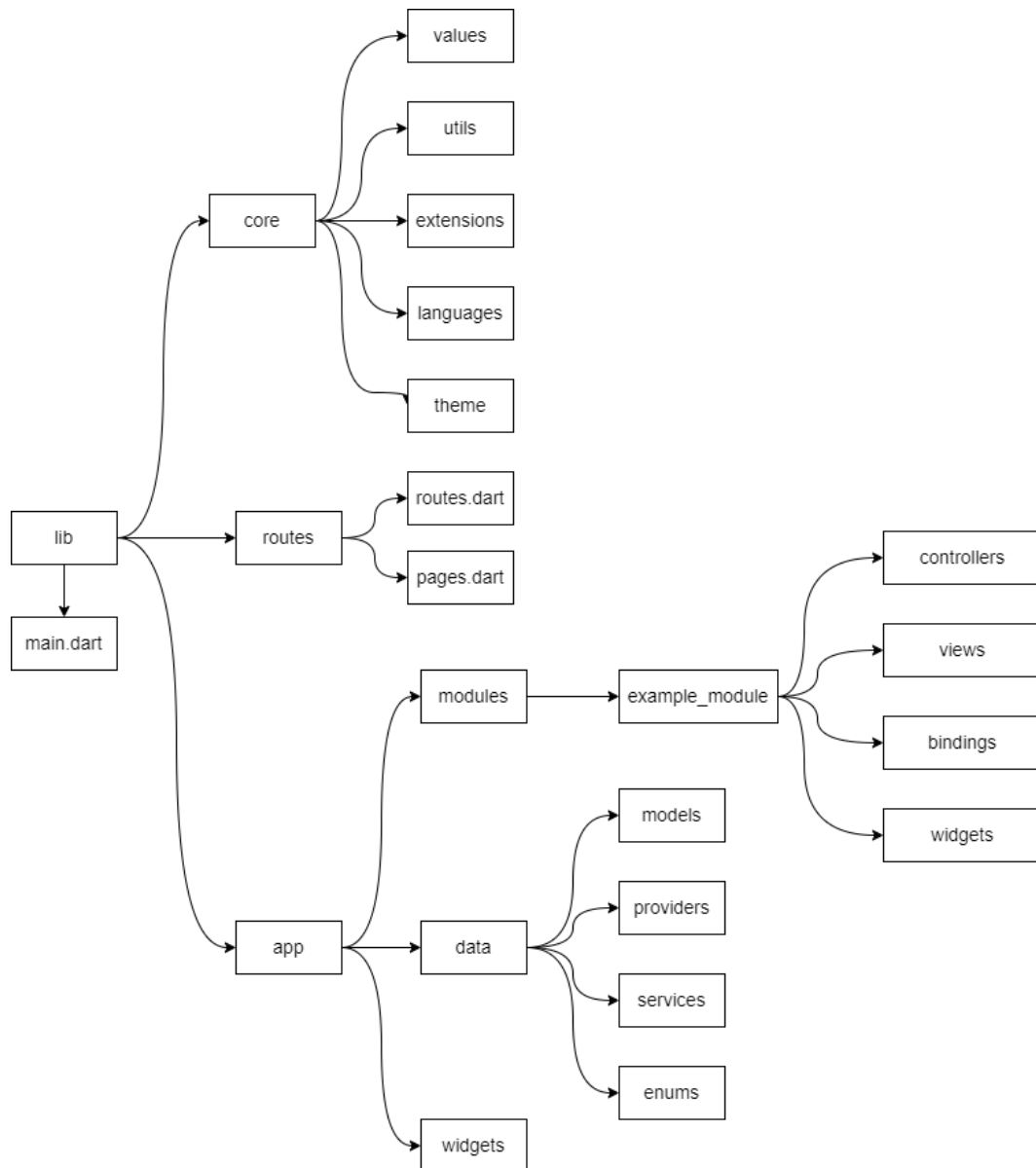
Platform UnilaHub (*Research Apps*) yang dikembangkan pada penelitian sebelumnya dibuat dengan menggunakan *framework* Flutter versi 2. Flutter versi 2 merupakan versi Flutter yang dirilis pada tahun 2021 sedangkan saat ini versi Flutter yang terbaru adalah Flutter versi 3. Migrasi kode program menggunakan

Flutter versi 3 menjadi salah satu tahapan penelitian yang akan dilakukan pada penelitian ini. Beberapa alasan yang dijadikan landasan dalam melakukan kegiatan migrasi kode program ke dalam Flutter versi 3 adalah sebagai berikut.

- a) Menghindari penggunaan *library* dan *built-in function* yang telah *deprecated*.
- b) Memanfaatkan fitur *Null Safety* yang tersedia di dalam *framework* Flutter untuk membantu pengembang mengatasi masalah umum terkait kesalahan penggunaan nilai *null* dalam kode program sehingga mengurangi resiko pada saat *runtime*.
- c) Memudahkan kegiatan pemeliharaan aplikasi dan pengembangan selanjutnya karena versi Flutter yang digunakan merupakan versi Flutter yang masih dianggap baru dalam perilisannya.

Penulisan kode program pada pengembangan *platform* UnilaHub (*Research Apps*) dalam penelitian sebelumnya masih terbilang sebagai *Spaghetti Code* yaitu kode program yang tidak terstruktur dan sulit untuk dipelihara serta banyak kode program yang ditulis secara berulang. Selain itu, pengelolaan *file* dan *folder* kode program pada pengembangan sebelumnya tidak mengikuti pola struktur tertentu seperti MVC, MVVM, atau yang lainnya.

Berdasarkan penjelasan tersebut, pada kegiatan migrasi kode program ke dalam Flutter versi 3 dilakukan juga *refactoring* kode program dan struktur direktori serta *file*. *Refactoring* kode program dilakukan dengan menerapkan *library* GetX untuk memudahkan dalam manajemen *state*, *dependency injection*, dan *navigation*. Perubahan struktur direktori dilakukan dengan menggunakan pola GetX Pattern. GetX Pattern merupakan pola struktur direktori yang mirip dengan pendekatan MVC (*Model*, *View*, *Controller*) yaitu memisahkan kode program untuk berdasarkan kategorinya, seperti kode program khusus untuk *view*, *controller*, *model*, *service*, dan yang lainnya. Berikut adalah rancangan struktur direktori kode program yang akan diimplementasikan dalam penelitian ini.



Gambar 6. Rancangan Struktur Direktori Kode Program.

b. Redesign Tampilan

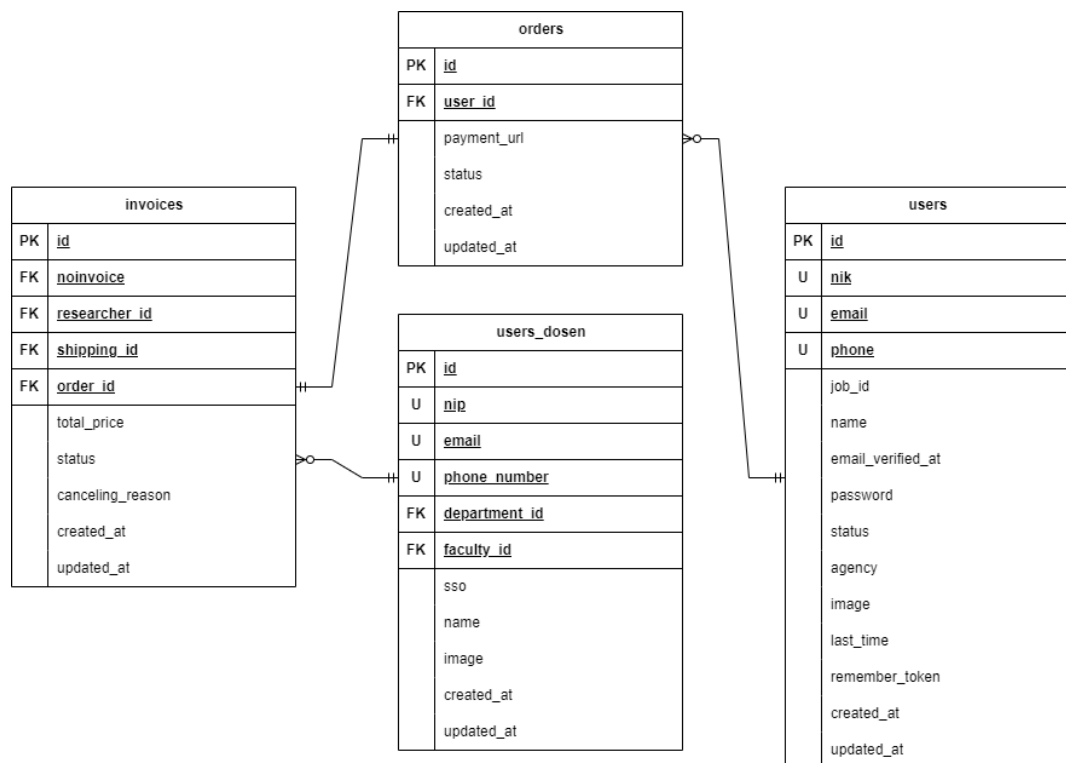
Redesign tampilan dilakukan dengan tujuan untuk membuat tampilan menjadi lebih baru dan mengadopsi penggunaan Material Design 3 yang secara tidak langsung tertanam pada Flutter versi 3. Perubahan tampilan yang dilakukan bukan merupakan perubahan secara menyeluruh. Namun, perubahan tampilan dilakukan pada beberapa bagian tampilan yang dirasa perlu untuk diperbarui. *Redesign* tampilan yang dilakukan hanya bergantung pada kepuasan pengguna tanpa menggunakan paradigma atau pendekatan apapun.

c. Rancangan Penambahan Fitur Pembayaran Pada *Platform UnilaHub* (*Public Apps*)

Penambahan fitur pembayaran dilakukan dengan mengimplementasikan *payment gateway* ke dalam aplikasi dengan tujuan untuk memudahkan pengguna baik pembeli maupun penjual dalam melakukan kegiatan transaksi. Penjual dan pembeli tidak lagi harus menunggu konfirmasi atau verifikasi pembayaran yang dilakukan secara manual oleh admin. Implementasi *payment gateway* mempercepat kegiatan jual beli yang dilakukan di dalam aplikasi. Halaman pembayaran akan muncul ketika pengguna telah selesai mengisi alamat dan data pengiriman produk yang dipesan.

a) *Logical Data Model* Fitur Pembayaran

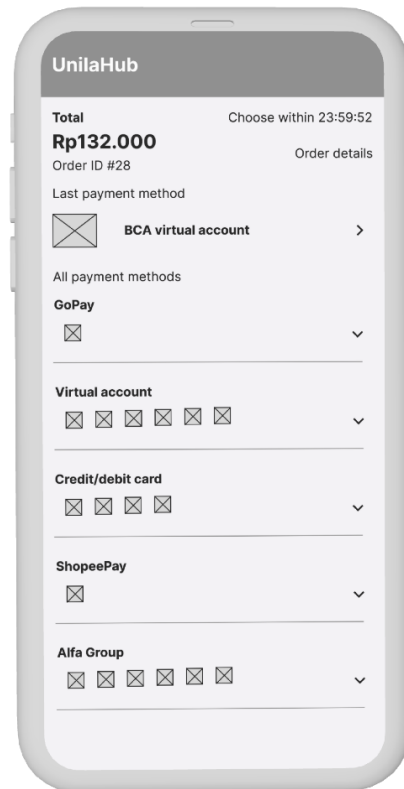
Berikut adalah *Logical Data Model* terkait dengan fitur pembayaran yang akan diimplementasikan pada aplikasi UnilaHub.



Gambar 7. *Logical Data Model* Fitur Pembayaran.

b) Rancangan Tampilan *List Metode Pembayaran*

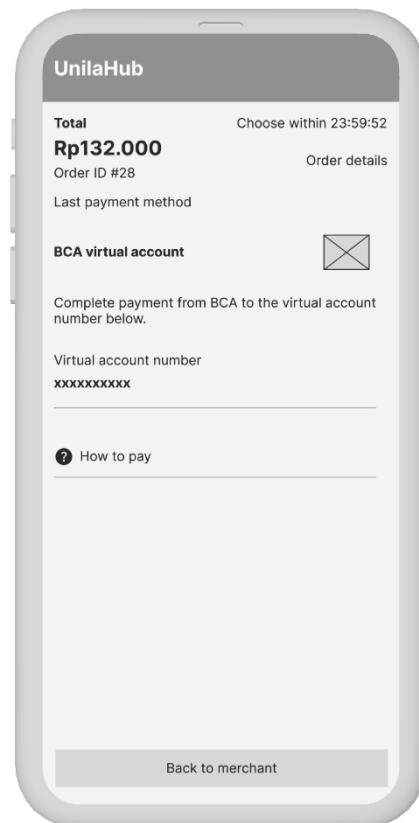
Halaman *list* metode pembayaran muncul ketika pengguna telah selesai mengisi data alamat dan informasi terkait pemesanan produk. Halaman ini berisikan beberapa metode yang dapat dipilih pengguna untuk melakukan pembayaran.



Gambar 8. Rancangan Tampilan *List Metode Pembayaran*.

c) Rancangan Tampilan Detail Metode Pembayaran

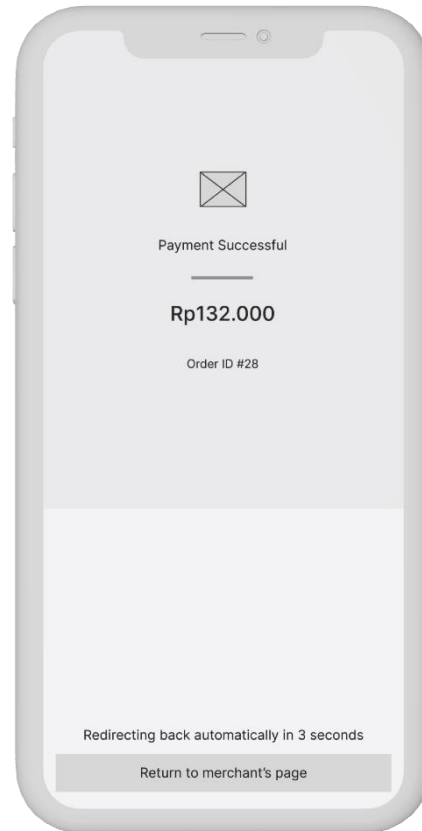
Halaman detail metode pembayaran muncul ketika pengguna memilih salah satu metode pembayaran yang tersedia pada halaman *list* metode pembayaran. Halaman detail metode pembayaran akan memberikan informasi lanjut mengenai cara untuk melakukan pembayaran menggunakan metode pembayaran yang dipilih.



Gambar 9. Rancangan Tampilan Detail Metode Pembayaran.

d) Rancangan Tampilan Pembayaran Berhasil

Halaman informasi yang menampilkan status pembayaran akan muncul secara otomatis ketika pengguna telah melakukan pembayaran sesuai dengan metode pembayaran yang dipilih.



Gambar 10. Rancangan Tampilan Halaman Pembayaran Berhasil.

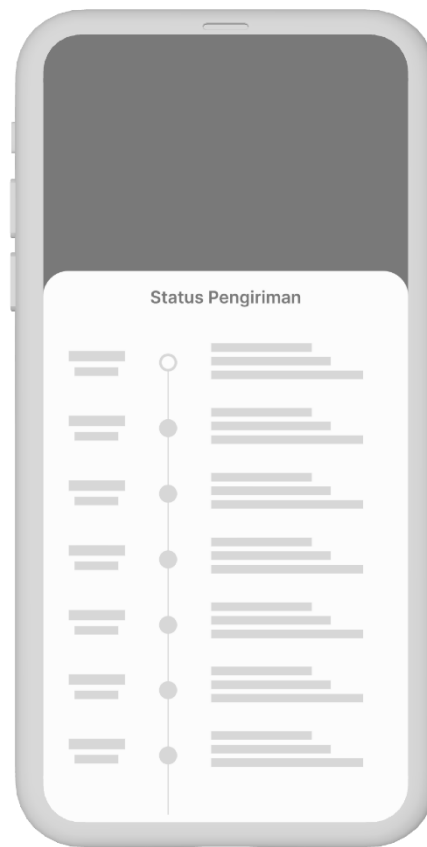
d. Rancangan Penambahan Fitur Pelacakan Pengiriman Pada Platform UnilaHub (*Public Apps*) dan UnilaHub (*Research Apps*)

Penambahan fitur pelacakan pengiriman bertujuan untuk memberikan informasi mengenai informasi status pengiriman sekaligus melacak keberadaan paket yang sedang dikirimkan. Fitur pelacakan pengiriman diharapkan dapat memberikan kenyamanan bagi pengguna baik pembeli maupun penjual dalam melakukan kegiatan jual beli di dalam aplikasi. Pelacakan pengiriman dapat diakses pada aplikasi di dalam detail pesanan. Fitur pelacakan pengiriman hanya akan tersedia di dalam detail pesanan yang berstatus “dikirim”, “diterima/selesai”, dan

“dibatalkan”. Fitur pelacakan pengiriman akan disesuaikan dengan jenis atau kategori produk. Produk yang berbentuk fisik akan terdapat fitur pelacakan pengiriman sedangkan produk yang berupa jasa maupun digital seperti *software* dan sejenisnya tidak memiliki fitur pelacakan pengiriman.

a) Rancangan Tampilan Pelacakan Pengiriman

Tampilan pelacakan pengiriman dapat diakses oleh pengguna melalui halaman detail pesanan dengan meng-klik tombol “Lacak Pengiriman”.



Gambar 11. Rancangan Tampilan Pelacakan Pengiriman.

b) *Logical Data Model* Fitur Pelacakan Pengiriman

Salah satu solusi alternatif dalam menerapkan fitur pelacakan pengiriman pada aplikasi UnilaHub adalah dengan memberikan atribut “has_shipping” pada tabel “categories” yang telah ada.

categories	
PK	<u>id</u>
	name
	icon
	created_at
	updated_at

Gambar 12. Tabel categories yang Telah Ada.

Gambar 12 merupakan bentuk tabel “categories” yang telah ada. Pada struktur tabel tersebut tidak terdapat atribut yang dapat dijadikan sebagai penanda sebuah produk dengan kategori tersebut memiliki fitur pengiriman atau tidak.

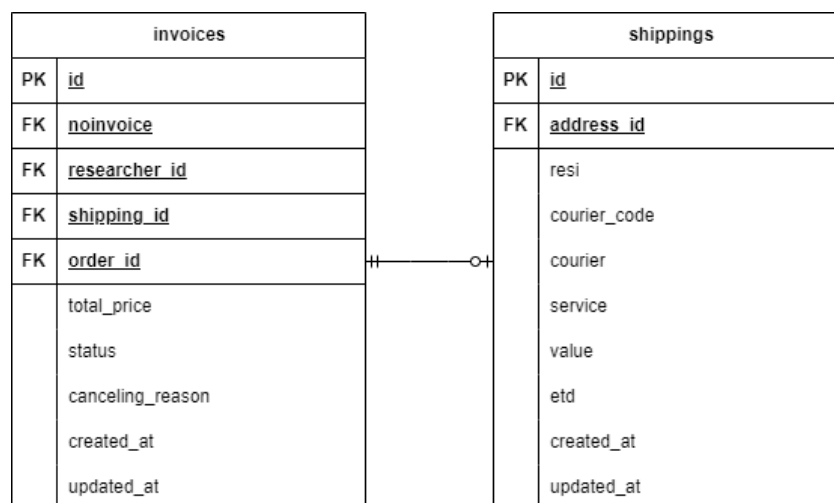
categories	
PK	<u>id</u>
	name
	icon
	has_shipping
	created_at
	updated_at

Gambar 13. Rancangan Perubahan Struktur Tabel categories.

Gambar 13 merupakan rancangan perubahan struktur tabel “categories” dengan menambahkan satu buah atribut “has_shipping” sebagai penanda sebuah produk dengan kategori tersebut memiliki fitur pengiriman atau tidak. Nilai “1” menandakan produk dengan kategori tersebut memiliki fitur pelacakan

pengiriman sedangkan nilai “0” menandakan produk tersebut tidak memiliki fitur pelacakan pengiriman.

Untuk mendukung fitur pelacakan pengiriman yang akan diimplementasikan, dibutuhkan beberapa tabel lain untuk menampung informasi terkait data pengiriman yang digunakan. Berikut adalah *Logical Data Model* fitur pengiriman yang akan diimplementasikan.



Gambar 14. Logical Data Model Fitur Pelacakan Pengiriman.

3.4.3.3. Iterasi *Prototyping* (Pengembangan Prototipe, Evaluasi Prototipe, dan Perbaikan dan Modifikasi Prototipe)

Tahapan yang dilakukan setelah menganalisis kebutuhan pengguna pada metode *prototyping* adalah iterasi pengembangan prototipe. Iterasi pengembangan prototipe adalah sebuah proses pengembangan prototipe yang dilakukan secara berulang-ulang atau secara bertahap, di mana pada setiap iterasi akan dilakukan beberapa hal yaitu pembuatan atau pengembangan prototipe awal, evaluasi prototipe berdasarkan pada hasil uji dan umpan balik pengguna, dan perbaikan serta modifikasi terhadap prototipe yang ada.

a. Pengembangan Prototipe Awal

Pengembangan prototipe awal dilakukan ketika kebutuhan pengguna serta rancangan dari sistem atau aplikasi telah selesai dilakukan. Pengembangan prototipe awal melibatkan pembuatan model awal atau prototipe dari sistem atau aplikasi yang akan dikembangkan. Tujuan dari tahap ini adalah untuk menyajikan aplikasi awal yang memenuhi fungsionalitas awal dari sistem kepada pengguna secara cepat.

b. Evaluasi dan Umpan Balik

Tahap evaluasi dan umpan balik didalam iterasi pengembangan prototipe merupakan salah satu tahapan utama di mana prototipe sistem akan dievaluasi oleh pengguna untuk mendapatkan umpan balik. Berdasarkan evaluasi dan umpan balik yang diberikan pengguna, tim pengembang akan mengumpulkan serta menganalisis informasi yang ada untuk melakukan perbaikan terhadap prototipe selanjutnya. Tahap evaluasi dan umpan balik berguna untuk memastikan bahwa aplikasi yang dikembangkan sesuai dengan kebutuhan dan harapan pengguna.

c. Perbaikan dan Modifikasi Prototipe

Perbaikan dan modifikasi prototipe dilakukan berdasarkan pada hasil evaluasi dan umpan balik yang diberikan oleh pengguna. Perbaikan dan modifikasi prototipe dapat mencakup beberapa aspek seperti penambahan fitur kecil, peningkatan antarmuka pengguna atau perubahan pada fungsionalitas dasar. Pengembang bekerja untuk memastikan bahwa setiap iterasi membawa perbaikan tambahan, mengarahkan prototipe lebih dekat ke solusi yang diinginkan.

3.4.3.4. Produk Akhir

Produk akhir merupakan aplikasi yang telah melewati serangkaian iterasi dan evaluasi, dan telah disempurnakan berdasarkan umpan balik yang diberikan oleh

pengguna. Produk akhir mencerminkan hasil dari upaya pengembang, umpan balik dari pengguna, dan kesiapan untuk memberikan solusi yang memenuhi kebutuhan pengguna.

3.4.4. Pembuatan Laporan

Tahap akhir pada penelitian ini adalah pembuatan laporan. Pembuatan laporan dilakukan dengan tujuan untuk mendokumentasikan penelitian yang dilakukan serta dengan harapan dapat dijadikan bahan referensi bagi penelitian yang akan datang.

DAFTAR PUSTAKA

- Al Alawi, A. (2022). Pengembangan Aplikasi Mobile *E-Commerce* Produk Riset Universitas Lampung Menggunakan Metode Scrum (*Public Apps*).
- Eviani, I., & Hidayat, Y. R. (2021). Pengaruh Sistem Pelacakan Online dan Ketepatan Waktu Pengiriman Terhadap Kepuasan Pelanggan (Studi Kasus J&T Express Kota Baru Bekasi). *Jurnal Manajemen Logistik*, 1(1), 11-19.
- Flutter. (2022). Flutter 3.0.0 release notes. diambil dari docs.flutter.dev: <https://docs.flutter.dev/release/release-notes/release-notes-3.0.0>
- Flutter. (2023). Flutter. diambil dari flutter.dev: <https://flutter.dev/>
- Get. (2023). Get. diambil dari pub.dev: <https://pub.dev/packages/get>
- GetX Pattern. (2023). GetX Pattern. diambil dari: https://kauemurakami.github.io/getx_pattern/
- Jan, S. R., Shah, S. T. U., Johar, Z. U., Shah, Y., & Khan, F. (2016). An innovative approach to investigate various software testing techniques and strategies. *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, Print ISSN, 23951990.
- Listianto, K. F., Fauzi, R. I., Irviani, R., Kasmi, K., & Garaika, G. (2017). Aplikasi E-Commerce Berbasis Web Mobile Pada Industri Konveksi Seragam Drumband Di Pekon Klaten Gadingrejo Kabupaten Pringsewu. *J. TAM (Technol. Accept. Model)*, 8(2), 146-152.
- Midtrans. (2023). Midtrans. diambil dari midtrans.com: <https://midtrans.com/>

- Ningsih, T. I., & Sanjaya, C. B. (2023). Aplikasi Pembayaran Sekolah Menggunakan Payment Gateway Tripay Berbasis Android. *Kohesi: Jurnal Sains dan Teknologi*, 1(1), 71-80.
- Nugraha, W., & Syarif, M. (2018). Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Penghitungan Volume Dan Cost Penjualan Minuman Berbasis Website. *JUSIM (Jurnal Sistem Informasi Musirawas)*, 3(2), 94-101.
- Nursaid, F. F., Brata, A. H., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus: Toko Uda Fajri). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(1), 46-55.
- Permana, F. M. (2022). Hilirisasi Produk Penelitian Universitas Lampung Menggunakan Platform E-Commerce Berbasis Mobile (Research Apps).
- Puspitasari, T. M. M., & Maulina, D. (2019). Implementasi payment gateway menggunakan midtrans pada marketplace travnesia. com. *Mobile and Forensics*, 1(1), 22-29.
- Putra, D. A., Sasmita, A., Made, G., & Wiranatha, A. K. A. C. (2020). E-Commerce marketplace petshop menggunakan integrasi rajaongkir API dan iPaymu Payment Gateway API. *Jurnal Ilmiah Teknologi dan Komputer*, 1(1), 13-22.
- RajaOngkir. (2023). RajaOngkir. diambil dari rajaongkir.com: <https://rajaongkir.com/>
- Sulaksana, E., Ichwani, A., Anwar, N., & Setiawati, P. (2023). Aplikasi E-Commerce Pakaian Berbasis Android Menggunakan Metode Prototype. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 7(2), 307-318.
- Syaputra, R., & Ganda, Y. P. W. (2019). Happy Flutter: Membuat Aplikasi Andorid dan iOS dengan Mudah menggunakan Flutter-UDACODING. *Udacoding*.