

**PEMBANGUNAN APLIKASI TOKO BUKU *ONLINE* BERBASIS
MOBILE MENGGUNAKAN FLUTTER**

**(Studi Kasus : Balai Besar Pengujian Mineral dan Batubara
tekMIRA)**

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan Program Strata 1,
di Program Studi Teknik Informatika, Universitas Pasundan

Oleh :

Herdy Sya'banul Hakim

Nrp. 22.304.0164



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PASUNDAN BANDUNG

2026

LEMBAR PENGESAHAN LAPORAN TUGAS AKHIR

Telah disetujui dan disahkan Laporan Tugas Akhir, dari :

Nama : Herdy Sya'banul Hakim
Nrp : 22.304.0164

Dengan Judul :

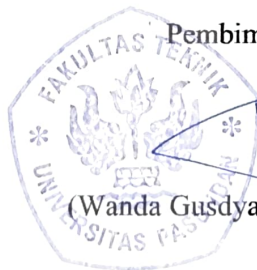
**“PEMBANGUNAN APLIKASI TOKO BUKU *ONLINE* BERBASIS *MOBILE*
MENGUNAKAN FLUTTER**

(Studi Kasus : Balai Besar Pengujian Mineral dan Batubara tekMIRA)”

Bandung, 22 September 2026

Menyetujui,

Pembimbing Utama



(Wanda Gusdya Purnama, ST., MT)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung.

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

Bandung, 22 September 2026

Yang membuat pernyataan,

Herdy Sya'banul Hakim

NRP. 22.304.0164

ABSTRAK

Balai Besar Pengujian Mineral dan Batubara tekMIRA merupakan instansi yang mengelola publikasi teknis berupa buku mengenai teknologi pertambangan, pengolahan, dan pemanfaatan mineral dan batu bara. Proses distribusi dan penjualan publikasi masih dilakukan melalui mekanisme konvensional, sedangkan pencatatan transaksi dilakukan secara terpisah sehingga dapat menyebabkan keterbatasan jangkauan distribusi, ketidakcocokan data, dan keterlambatan pelaporan penjualan. Berdasarkan permasalahan tersebut, tugas akhir ini bertujuan untuk membangun aplikasi toko buku *online* berbasis *mobile* menggunakan Flutter yang dapat mendukung proses penjualan, pencatatan transaksi, dan pelaporan penjualan secara terpusat.

Pembangunan aplikasi dilakukan melalui tahapan identifikasi masalah, pengumpulan data melalui studi literatur dan wawancara, serta pembangunan perangkat lunak secara iteratif yang meliputi *planning*, *design*, *develop*, dan *testing*. Pada tahap *design* dilakukan analisis kebutuhan, pemodelan sistem menggunakan UML, perancangan arsitektur, dan perancangan antarmuka pengguna, sedangkan tahap *develop* dilakukan dengan menggunakan Flutter dan Dart serta integrasi dengan sistem *backend* melalui REST API. Aplikasi yang dibangun mencakup fungsi utama untuk *customer*, seperti melihat katalog dan detail buku, mengelola keranjang dan alamat pengiriman, melakukan pemesanan, mengunggah bukti pembayaran, serta melihat riwayat pesanan dan notifikasi, serta fungsi admin untuk melihat *dashboard*, mengelola pesanan, mencatat penjualan tunai, dan mengelola data buku.

Hasil tugas akhir berupa Aplikasi Toko Buku Online tekMIRA berbasis *mobile* yang telah diimplementasikan menggunakan Flutter dan terintegrasi dengan sistem *backend* melalui REST API. Aplikasi dapat mendukung proses penjualan publikasi secara terpusat, mulai dari penyajian katalog, pengelolaan keranjang, pemesanan, pembayaran, hingga pemantauan status pesanan, serta menyediakan fungsi pengelolaan penjualan pada sisi admin. Berdasarkan hasil pengujian fungsional menggunakan metode *black-box testing*, fungsi-fungsi yang diuji dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan sehingga aplikasi dapat digunakan sebagai sarana pendukung proses penjualan dan pencatatan publikasi tekMIRA.

Kata kunci: aplikasi *mobile*, toko buku *online*, Flutter, REST API, *black-box testing*, tekMIRA

ABSTRACT

The Center for Mineral and Coal Testing tekMIRA is an institution that manages technical publications in the form of books related to mining technology, mineral processing, and mineral and coal utilization. The distribution and sales of these publications are still conducted through conventional mechanisms, while transaction records are maintained separately, which may result in limited distribution reach, data discrepancies, and delays in sales reporting. Based on these problems, this final project aims to build a mobile-based online bookstore application using Flutter to support centralized sales, transaction recording, and sales reporting.

The application was developed through problem identification, data collection using literature study and interviews, and iterative software development consisting of planning, design, development, and testing. The design stage included requirements analysis, system modeling using UML, architectural design, and user interface design, while the development stage used Flutter and Dart and integrated the application with the backend system through REST API. The developed application provides main functions for customers, including viewing the book catalog and details, managing the shopping cart and shipping addresses, placing orders, uploading payment proof, and viewing order history and notifications, as well as administrative functions for viewing the dashboard, managing orders, recording cash sales, and managing book data.

The result of this final project is the tekMIRA Online Bookstore Application, a mobile application implemented using Flutter and integrated with the backend system through REST API. The application supports a centralized publication sales process, including catalog presentation, shopping cart management, ordering, payment, and order status monitoring, while also providing sales management functions for administrators. Based on functional testing using the black-box testing method, the tested functions operated according to the specified requirements, indicating that the application can be used as a supporting tool for the sales and transaction recording processes of tekMIRA publications.

Keywords: mobile application, online bookstore, Flutter, REST API, black-box testing, tekMIRA.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Pembangunan Aplikasi Toko Buku Online Berbasis Mobile Menggunakan Flutter (Studi Kasus: Balai Besar Pengujian Mineral dan Batubara tekMIRA)”.

Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pasundan Bandung. Tugas Akhir ini membahas pembangunan aplikasi toko buku *online* berbasis *mobile* yang ditujukan untuk membantu proses penjualan publikasi teknis pada Balai Besar Pengujian Mineral dan Batubara tekMIRA.

Dalam penyusunan laporan dan pengerjaan Tugas Akhir ini, penulis memperoleh banyak bantuan, arahan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak Wanda Gusdya Purnama, ST., MT., selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan masukan selama penyusunan Tugas Akhir ini.
2. Seluruh dosen Program Studi Teknik Informatika Universitas Pasundan yang telah memberikan ilmu dan pengalaman selama masa perkuliahan.
3. Pihak Balai Besar Pengujian Mineral dan Batubara tekMIRA yang telah memberikan kesempatan dan informasi yang dibutuhkan dalam penyusunan Tugas Akhir ini.
4. Orang tua dan keluarga yang selalu memberikan doa, dukungan, dan motivasi kepada penulis.
5. Teman-teman seperjuangan, yaitu Dicky Dwi Dermawan, Genta Sena Irasta, Ilman Hanifa, Muhammad Kholis Kamil, Muhammad Rinaldi Agus Pratama, Muhammad Rizki Maulana, Miftahul Huda, Ryan Hidayat, Mustika Weni, Annisa Septiyani, dan Yesi Dedehidayati, atas kebersamaan, bantuan, dukungan, dan berbagai pengalaman yang telah dibagikan selama masa perkuliahan, sejak awal semester hingga akhir masa studi.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang membutuhkan.

Bandung, 22 September 2026

Herdy Sya'banul Hakim

DAFTAR ISI

LEMBAR PENGESAHAN LAPORAN TUGAS AKHIR	i
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR ISTILAH.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1-1
1.1 Latar Belakang Masalah	1-1
1.2 Identifikasi Masalah	1-2
1.3 Tujuan Tugas Akhir	1-2
1.4 Lingkup Tugas Akhir	1-3
1.5 Metodologi Pengerjaan Tugas Akhir	1-3
1.6 Sistematika Penulisan Laporan Tugas Akhir.....	1-4
BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU	2-1
2.1 Teori Pendukung	2-1
2.1.1 <i>E-Commerce</i>	2-1
2.1.2 <i>Mobile Application</i>	2-2
2.1.3 <i>Framework</i> Flutter.....	2-3
2.1.4 Rekayasa Perangkat Lunak	2-5
2.1.5 Pengembangan Perangkat Lunak <i>Agile</i>	2-6
2.1.6 Firebase	2-7
2.1.7 Firebase Cloud Messaging (FCM)	2-7
2.2 Penelitian Terdahulu	2-8
BAB 3 SKEMA PENELITIAN.....	3-1
3.1 Rancangan Penelitian.....	3-1
3.2 Perumusan Masalah.....	3-2
3.2.1 Analisis Sebab Akibat.....	3-3
3.2.2 Solusi Masalah.....	3-4
3.3 Kerangka Berpikir Teoritis	3-5
3.3.1 Gambaran Produk Tugas Akhir.....	3-5
3.3.2 Skema Analisis Teori.....	3-6

3.4 Profile Penelitian	3-8
BAB 4 ANALISIS DAN PERANCANGAN.....	4-1
4.1 Analisis Kebutuhan	4-1
4.1.1 Analisis <i>Stakeholder</i>	4-1
4.1.2 Analisis Proses Bisnis Berjalan.....	4-1
4.1.3 Analisis Kebutuhan Fungsional.....	4-2
4.1.4 Analisis Pengguna.....	4-3
4.2 Pemodelan berbasis Skenario	4-4
4.2.1 <i>Use case Diagram</i>	4-4
4.2.2 Skenario <i>Use case</i>	4-5
4.3 Perancangan Kelas	4-11
4.3.1 Mengidentifikasi Kelas-Kelas Analisis.....	4-11
4.3.2 Menentukan Atribut-Atribut.....	4-12
4.3.3 Mendefinisikan Operasi-Operasi.....	4-13
4.3.4 Perancangan Kelas	4-16
4.3.5 Diagram Sekuens.....	4-17
4.4 Perancangan Perangkat Lunak	4-22
4.4.1 Perancangan Data	4-22
4.4.2 Perancangan Arsitektural	4-25
4.4.3 Perancangan Antarmuka Pengguna	4-26
BAB 5 IMPLEMENTASI DAN PENGUJIAN	5-1
5.1 Konstruksi Perangkat Lunak.....	5-1
5.1.1 Kebutuhan Perangkat Lunak	5-1
5.2 Pengkodean.....	5-2
5.2.1 Tahap Pengkodean Aplikasi Mobile	5-2
5.2.2 Implementasi Antarmuka Pengguna.....	5-2
5.3 Pengujian.....	5-6
BAB 6 PENUTUP.....	6-1
6.1 Kesimpulan	6-1
6.2 Saran.....	6-1
DAFTAR PUSTAKA.....	xiv
LAMPIRAN A BERITA ACARA WAWANCARA	A-1

DAFTAR ISTILAH

Tabel 1 Daftar Istilah

No.	Istilah	Keterangan
1.	<i>Electronic commerce (EC / e-commerce)</i>	Perdagangan elektronik yang menggunakan internet atau intranet untuk membeli, menjual, memindahkan, atau memperdagangkan data, barang, maupun jasa [TUR15].
2.	<i>Mobile commerce (m-commerce)</i>	Pelaksanaan <i>e-commerce</i> menggunakan perangkat <i>mobile</i> dan jaringan nirkabel [TUR15].
3.	<i>Pure EC</i>	Bentuk <i>e-commerce</i> ketika proses pemesanan, pembayaran, pemenuhan pesanan, dan pengiriman dilakukan secara digital [TUR15].
4.	<i>Partial EC</i>	Bentuk <i>e-commerce</i> ketika paling sedikit satu aktivitas transaksi masih dilakukan secara fisik [TUR15].
5.	<i>Business-to-Consumer (B2C)</i>	Model transaksi ritel produk atau jasa dari pelaku bisnis kepada konsumen individual [TUR15].
6.	<i>E-tailing</i>	Aktivitas penjualan ritel secara elektronik, khususnya dalam model B2C [TUR15].
7.	<i>World wide web</i>	Sistem layanan web yang menjadi salah satu faktor perkembangan <i>electronic commerce</i> .
8.	Webstore	Toko daring yang digunakan untuk menampilkan informasi produk dan mendukung aktivitas penjualan secara elektronik [TUR15].
9.	E-catalog	Katalog elektronik yang menyajikan informasi produk dalam format digital [TUR15].
10.	<i>Electronic shopping cart</i>	Komponen perangkat lunak yang memungkinkan pelanggan mengumpulkan produk sebelum melakukan <i>checkout</i> [TUR15].
11.	<i>Checkout</i>	Proses akhir ketika pelanggan melanjutkan produk dalam keranjang belanja ke tahap pemesanan atau pembayaran.
12.	<i>Payment gateway</i>	Gerbang pembayaran yang digunakan untuk memproses transaksi pembayaran secara aman [TUR15].
13.	<i>Customer services</i>	Layanan pelanggan yang mencakup penyediaan informasi produk, bantuan transaksi, dan pengelolaan hubungan dengan pelanggan [TUR15].
14.	<i>Front end</i>	Bagian sistem yang menjadi tempat pengguna berinteraksi langsung dengan aplikasi [TUR15].
15.	<i>Back end</i>	Bagian sistem yang menangani pemenuhan pesanan, manajemen inventaris, dan pemrosesan data [TUR15].
16.	<i>Mobile application</i>	Perangkat lunak yang dirancang untuk dijalankan pada perangkat bergerak seperti <i>smartphone</i> dan <i>tablet</i> [WAS10].
17.	<i>Smartphone</i>	Perangkat telepon pintar yang dapat menjalankan aplikasi dan terhubung ke internet.
18.	<i>Tablet</i>	Perangkat bergerak berlayar sentuh yang dapat digunakan untuk menjalankan aplikasi.
19.	<i>Native</i>	Pendekatan pengembangan aplikasi yang dibuat khusus untuk satu platform tertentu [WAS10].
20.	Native app	Aplikasi yang dikembangkan secara khusus untuk satu platform, seperti Android atau iOS [WAS10].
21.	Mobile web app	Aplikasi yang diakses melalui <i>browser</i> pada perangkat <i>mobile</i> [WAS10].

No.	Istilah	Keterangan
22.	<i>Hybrid app</i>	Aplikasi yang menggabungkan teknologi web dan kemampuan <i>native</i> pada perangkat [WAS10].
23.	<i>Cross-platform</i>	Pendekatan pengembangan aplikasi agar dapat berjalan pada lebih dari satu platform [WAS10].
24.	Single codebase	Satu basis kode yang digunakan untuk membangun aplikasi pada lebih dari satu platform.
25.	User interface (UI)	Antarmuka pengguna, yaitu tampilan dan sarana interaksi antara pengguna dengan aplikasi [WAS10].
26.	<i>Client-server</i>	Arsitektur sistem yang membagi peran antara aplikasi klien sebagai pengirim permintaan dan server sebagai penyedia layanan [WAS10].
27.	<i>Client</i>	Aplikasi atau perangkat pengguna yang mengirim permintaan layanan kepada server.
28.	<i>Server</i>	Sistem yang menerima permintaan, memproses data, dan memberikan respons kepada klien.
29.	<i>Application Programming Interface (API)</i>	Sekumpulan fitur, aturan, dan antarmuka logis yang memungkinkan perangkat lunak lain berinteraksi dengan suatu aplikasi.
30.	REST API	Antarmuka yang digunakan aplikasi untuk berkomunikasi dengan sistem <i>backend</i> menggunakan prinsip REST.
31.	Android	Platform utama yang menjadi sasaran pengembangan aplikasi dalam tugas akhir ini.
32.	iOS	Sistem operasi perangkat bergerak yang dikembangkan oleh Apple.
33.	Platform	Lingkungan perangkat lunak atau sistem operasi tempat aplikasi dijalankan.
34.	Browser	Perangkat lunak yang digunakan untuk mengakses halaman atau aplikasi berbasis web.
35.	Framework	Kerangka kerja yang menyediakan struktur, komponen, dan fasilitas untuk membangun perangkat lunak.
36.	<i>Flutter</i>	<i>Framework</i> yang digunakan untuk membangun aplikasi <i>mobile</i> pada tugas akhir ini [WIN20].
37.	<i>Dart</i>	Bahasa pemrograman yang digunakan dalam pengembangan aplikasi Flutter [WIN20].
38.	<i>Software Development Kit (SDK)</i>	Kumpulan alat bantu yang digunakan untuk mengembangkan aplikasi pada platform atau <i>framework</i> tertentu.
39.	<i>Rendering engine</i>	Komponen yang bertugas menggambar atau menampilkan elemen antarmuka aplikasi [WIN20].
40.	<i>Testing framework</i>	Kerangka kerja yang mendukung proses pengujian perangkat lunak [WIN20].
41.	<i>Router</i>	Komponen yang mengatur perpindahan halaman atau rute dalam aplikasi [WIN20].
42.	<i>JavaScript bridge</i>	Lapisan penghubung antara kode JavaScript dan kode <i>native</i> perangkat [WIN20].
43.	ARM	Arsitektur prosesor yang umum digunakan pada perangkat <i>mobile</i> .
44.	<i>Ahead-of-time (AOT)</i>	Teknik kompilasi kode sebelum aplikasi dijalankan sehingga menghasilkan kode <i>native</i> untuk kebutuhan produksi [WIN20].
45.	<i>Just-in-time (JIT)</i>	Teknik kompilasi saat aplikasi dijalankan atau saat proses pengembangan berlangsung [WIN20].

No.	Istilah	Keterangan
46.	<i>Hot reload</i>	Fitur Flutter yang memungkinkan perubahan kode langsung terlihat tanpa menjalankan ulang aplikasi secara penuh [WIN20].
47.	<i>Widget</i>	Komponen dasar pembentuk antarmuka pada Flutter [WIN20].
48.	<i>Widget tree</i>	Struktur hierarki susunan <i>widget</i> dalam aplikasi Flutter [WIN20].
49.	<i>Document Object Model (DOM)</i>	Struktur representasi elemen pada pengembangan web yang digunakan sebagai pembanding dengan struktur <i>widget tree</i> .
50.	<i>StatelessWidget</i>	Jenis <i>widget</i> Flutter yang tidak memiliki kondisi internal yang berubah [WIN20].
51.	<i>StatefulWidget</i>	Jenis <i>widget</i> Flutter yang memiliki kondisi internal yang dapat berubah [WIN20].
52.	<i>State</i>	Kondisi atau data internal yang memengaruhi tampilan dan perilaku komponen aplikasi [WIN20].
53.	<i>setState</i>	Metode pada Flutter yang digunakan untuk memperbarui tampilan ketika <i>state</i> berubah [WIN20].
54.	<i>Software engineering</i>	Rekayasa perangkat lunak, yaitu penerapan pendekatan sistematis, terdisiplin, dan terukur terhadap pengembangan, pengoperasian, dan pemeliharaan perangkat lunak [PRE10].
55.	<i>Agile</i>	Pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi, dan penyerahan perangkat lunak secara bertahap [PRE10].
56.	<i>Agile software development</i>	Paradigma pengembangan perangkat lunak yang menekankan respons terhadap perubahan dan pengiriman perangkat lunak secara inkremental [PRE10].
57.	<i>Agile Manifesto</i>	Pernyataan nilai dan prinsip yang menjadi dasar pendekatan pengembangan perangkat lunak <i>agile</i> [PRE10].
58.	<i>Planning</i>	Tahap perencanaan dalam proses pembangunan perangkat lunak.
59.	<i>Design</i>	Tahap perancangan arsitektur, pemodelan sistem, antarmuka pengguna, dan alur navigasi aplikasi.
60.	<i>Develop</i>	Tahap implementasi atau pengkodean aplikasi berdasarkan rancangan yang telah disusun.
61.	<i>Testing</i>	Tahap pengujian untuk memverifikasi bahwa fungsi aplikasi berjalan sesuai kebutuhan.
62.	<i>Communication</i>	Aktivitas komunikasi dalam rekayasa perangkat lunak untuk memahami kebutuhan pengguna [PRE10].
63.	<i>Modeling</i>	Aktivitas pemodelan dalam rekayasa perangkat lunak untuk menggambarkan kebutuhan dan rancangan sistem [PRE10].
64.	<i>Construction</i>	Aktivitas pembangunan perangkat lunak yang mencakup pengkodean dan pengujian [PRE10].
65.	<i>Deployment</i>	Aktivitas penempatan perangkat lunak agar dapat digunakan oleh pengguna [PRE10].
66.	<i>Acceptance testing</i>	Pengujian penerimaan untuk menentukan apakah sistem telah memenuhi kriteria kebutuhan pengguna.
67.	<i>Usability</i>	Tingkat kegunaan sistem dalam mendukung pengguna mencapai tujuan secara efektif, efisien, dan memuaskan.
68.	<i>fishbone diagram</i>	Diagram sebab akibat yang digunakan untuk mengelompokkan kemungkinan penyebab suatu masalah.

No.	Istilah	Keterangan
69.	<i>Unified Modeling Language (UML)</i>	Bahasa pemodelan grafis yang digunakan untuk menggambarkan rancangan sistem.
70.	<i>Firebase</i>	Platform pengembangan aplikasi berbasis <i>cloud</i> yang menyediakan layanan pendukung pengembangan aplikasi [MOR17].
71.	Cloud	Infrastruktur komputasi berbasis internet yang menyediakan layanan penyimpanan, pemrosesan, atau pengiriman data.
72.	Firebase Cloud Messaging (FCM)	Layanan Firebase untuk mengirim pesan dan notifikasi ke aplikasi pengguna [FCM26].
73.	Notification message	Jenis pesan FCM yang umumnya ditampilkan sebagai notifikasi kepada pengguna [FCM26].
74.	<i>Data message</i>	Jenis pesan FCM yang muatannya diproses oleh logika aplikasi [FCM26].
75.	<i>Payload</i>	Muatan data yang dibawa dalam pesan atau permintaan sistem.
76.	<i>Firebase Admin SDK</i>	SDK sisi server yang digunakan untuk mengelola layanan Firebase dan mengirim pesan dari lingkungan terpercaya [FCM26].
77.	<i>Marketplace</i>	Kanal digital umum yang digunakan untuk memasarkan atau menjual produk melalui platform pihak ketiga.
78.	<i>Open booth</i>	Kegiatan membuka stan penjualan secara langsung pada pameran atau acara tertentu.
79.	<i>Focal point</i>	Pihak atau unit yang menjadi titik koordinasi utama dalam proses tertentu.
80.	<i>Online</i>	Kondisi ketika aktivitas atau layanan dilakukan melalui jaringan internet.
81.	<i>Real-time</i>	Kondisi ketika informasi diperbarui dan dapat dipantau secara langsung atau hampir langsung.
82.	<i>Code editor</i>	Perangkat lunak yang digunakan untuk menulis dan mengelola kode program.

DAFTAR TABEL

Tabel 1 Daftar Istilah	viii
Tabel 2. 1 Perbandingan Pendekatan Pengembangan Aplikasi Mobile	2-4
Tabel 2. 2 Penelitian Terdahulu	2-8
Tabel 3. 1 Alur Penyelesaian Tugas Akhir	3-1
Tabel 3. 2 Analisis Sebab Akibat	3-3
Tabel 3. 3 Solusi Masalah	3-4
Tabel 3. 4 Penjelasan Kerangka Solusi Masalah.....	3-4
Tabel 3. 5 Kerangka Pemikiran Teoritis	3-5
Tabel 3. 6 Detail Skema Analisis Teori	3-7
Tabel 4. 1 Analisis Stakeholder.....	4-1
Tabel 4. 2 Analisis Kebutuhan Fungsional	4-3
Tabel 4. 3 Analisis Pengguna	4-4
Tabel 4. 4 Deskripsi Use case Diagram	4-4
Tabel 4. 5 Deskripsi Relasi Use case	4-5
Tabel 4. 6 Use case Register Akun	4-5
Tabel 4. 7 Use case Melihat Katalog Buku	4-6
Tabel 4. 8 Use case Mengelola Keranjang.....	4-6
Tabel 4. 9 Use case Mengelola Alamat Pengiriman	4-7
Tabel 4. 10 Use case Melakukan Pesanan.....	4-7
Tabel 4. 11 Use case Melihat Riwayat Pesanan.....	4-8
Tabel 4. 12 Use case Melihat Dashboard.....	4-9
Tabel 4. 13 Use case Mengelola Pesanan.....	4-9
Tabel 4. 14 Use case Mengelola Data Buku	4-10
Tabel 4. 15 Kelas - Kelas Analisis	4-11
Tabel 4. 16 Atribut - Atribut Kelas	4-12
Tabel 4. 17 Operasi - Operasi Kelas	4-13
Tabel 4. 18 Perancangan Data.....	4-22
Tabel 5. 1 Spesifikasi Minimum Perangkat Keras Pengembangan.....	5-1
Tabel 5. 2 Perangkat Lunak yang Digunakan	5-1
Tabel 5. 3 Black Box Testing.....	5-7

DAFTAR GAMBAR

Gambar 1. 1 Metodologi Pengerjaan Tugas Akhir.....	1-3
Gambar 2. 1 Flutter	2-3
Gambar 2. 2 Agile Methodology.....	2-6
Gambar 2. 3 Firebase	2-7
Gambar 3. 1 Kerangka Permasalahan	3-3
Gambar 3. 2 Kerangka Solusi Masalah	3-4
Gambar 3. 3 Kerangka Pemikiran Teoritis.....	3-5
Gambar 3. 4 Skema Analisis Teori	3-7
Gambar 3. 5 Profile Penelitian	3-8
Gambar 4. 1 Diagram Aktivitas	4-2
Gambar 4. 2 Use Case Diagram	4-4
Gambar 4. 3 Diagram Kelas	4-17
Gambar 4. 4 Diagram Sekuens Melihat Katalog Buku	4-18
Gambar 4. 5 Gambar Sekuens Mengelola Keranjang	4-18
Gambar 4. 6 Diagram Sekuens Melakukan Pemesanan.....	4-19
Gambar 4. 7 Diagram Sekuens Melihat Dashboard	4-20
Gambar 4. 8 Diagram Sekuens Mengelola Pesanan.....	4-20
Gambar 4. 9 Diagram Sekuens Mengelola Buku	4-21
Gambar 4. 10 Diagram Sekuens Mencatat Penjualan Tunai atau Langsung	4-22
Gambar 4. 11 Perancangan Arsitektural	4-26
Gambar 4. 12 Rancangan Katalog Buku	4-26
Gambar 4. 13 Rancangan Halaman Keranjang	4-27
Gambar 4. 14 Rancangan Halaman Checkout.....	4-27
Gambar 4. 15 Rancangan Halaman Pembayaran	4-28
Gambar 4. 16 Rancangan Halaman Pesanan.....	4-28
Gambar 4. 17 Rancangan Halaman Dashboard Admin	4-29
Gambar 4. 18 Rancangan Halaman Manajemen Pesanan Admin.....	4-29
Gambar 4. 19 Rancangan Halaman Buku Admin	4-30
Gambar 5. 1 Struktur Folder Aplikasi Mobile	5-2
Gambar 5. 2 Implementasi Halaman Katalog dan Detail Buku	5-3
Gambar 5. 3 Implementasi Halaman Keranjang	5-3
Gambar 5. 4 Implementasi Halaman <i>Checkout</i> dan Pembayaran	5-4
Gambar 5. 5 Implementasi Halaman Pesanan.....	5-4
Gambar 5. 6 Implementasi Halaman Dashboard Admin	5-5
Gambar 5. 7 Implementasi Halaman Manajemen Pesanan	5-5
Gambar 5. 8 Implementasi Halaman Manajemen Buku Admin	5-6

DAFTAR LAMPIRAN

Lampiran A Berita Acara Wawancara	A-1
---	-----

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Perdagangan elektronik berbasis perangkat bergerak (*mobile commerce*) telah diakui sebagai infrastruktur distribusi barang dan jasa yang strategis, termasuk dalam sektor penerbitan dan penjualan buku. Pemerintah Indonesia melalui Perpres No.74 Tahun 2017 tentang Peta Jalan *e-Commerce* Nasional menetapkan bahwa transformasi digital pada rantai distribusi produk, termasuk produk pengetahuan berbasis cetak dan digital, merupakan prioritas pembangunan ekonomi berbasis inovasi. Dalam kerangka tersebut, institusi pemerintah maupun lembaga penelitian negara didorong untuk mengadopsi kanal distribusi digital guna memperluas jangkauan layanan publik dan meningkatkan efisiensi operasional.

Balai Besar Pengujian Mineral dan Batubara tekMira, sebagai unit pelaksana teknis di bawah Kementerian Energi dan Sumber Daya Mineral, mengelola dan menerbitkan berbagai publikasi teknis mengenai teknologi pertambangan, pengolahan, dan pemanfaatan mineral dan batubara di Indonesia yang merupakan hasil karya para ahli yang dimiliki oleh institusi. Hingga periode penelitian ini dilaksanakan, mekanisme distribusi dan penjualan publikasi tersebut masih bertumpu pada kanal konvensional, yakni melalui *open booth* di pameran, acara pertemuan yang diselenggarakan oleh Direktorat Jenderal Mineral dan batubara, serta pemasaran melalui media sosial dan *marketplace* umum. Berdasarkan data yang diperoleh melalui wawancara dengan pihak tekMIRA, tercatat 122 eksemplar buku berhasil terjual sepanjang tahun 2025 dengan rata-rata 10 buku per bulan, suatu angka yang mencerminkan keterbatasan jangkauan kanal distribusi yang berjalan. Selain itu, proses pencatatan transaksi dilakukan secara konvensional oleh bendahara penerimaan dan kemudian disinkronkan dengan Subkoordinator Hukum, Kerjasama, dan Informasi (HKI) selaku *focal point* penjualan, sehingga perbedaan metode pembayaran antara transfer bank dan pembayaran tunai di lokasi pameran kerap menimbulkan ketidakcocokan data antara kedua unit serta keterlambatan pembaruan informasi penjualan. Kondisi tersebut menyebabkan potensi diseminasi pengetahuan teknis kepada pemangku kepentingan industri tidak terealisasi secara optimal.

Sejumlah penelitian terdahulu telah mengupayakan digitalisasi penjualan buku melalui berbagai pendekatan teknologi. Bregaster dkk. mengembangkan sistem penjualan buku berbasis web untuk toko buku komersial swasta dengan pendekatan berorientasi objek, namun sistem yang dihasilkan terbatas pada platform desktop dan tidak mengakomodasi aksesibilitas melalui perangkat *mobile* [BRE22]. Pada ranah aplikasi *mobile*, Prasetya dkk. serta Hartati dan Al Fayyadh membangun aplikasi penjualan buku berbasis Android masing-masing menggunakan metode RAD dan pendekatan berorientasi objek [PRA25, HAR25], meskipun keduanya dikembangkan secara eksklusif untuk satu platform sehingga tidak menjangkau pengguna ekosistem iOS, suatu keterbatasan yang bahkan diakui secara eksplisit oleh Hartati dan Al Fayyadh sebagai rekomendasi pengembangan lanjutan [HAR25]. Selain keterbatasan

platform, penelitian-penelitian tersebut berfokus pada entitas bisnis swasta dengan produk buku konsumtif untuk pasar umum. Sementara itu, penelitian ini berfokus pada pembangunan sistem distribusi dan penjualan publikasi teknis ilmiah milik lembaga pemerintahan dengan segmen pengguna spesifik, yang membutuhkan jangkauan lintas ekosistem perangkat dan efisiensi pengembangan sesuai keterbatasan anggaran institusi publik.

Pemilihan platform *mobile* sebagai basis pengembangan turut mempertimbangkan kebutuhan fungsional aplikasi, khususnya penyampaian notifikasi terkait aktivitas pembayaran dan pesanan kepada pengguna melalui Firebase Cloud Messaging (FCM). Karakteristik aplikasi *mobile* memungkinkan notifikasi diterima secara langsung melalui perangkat pengguna sehingga informasi perubahan status pesanan dapat disampaikan secara lebih cepat dan mudah diakses. Kebutuhan tersebut mendukung penggunaan aplikasi *mobile* sebagai kanal penjualan yang tidak hanya menyediakan akses katalog dan pemesanan, tetapi juga pemantauan proses transaksi oleh pengguna.

Di antara berbagai pendekatan pengembangan aplikasi *mobile*, pendekatan *cross-platform* menggunakan *framework* Flutter dipilih karena memungkinkan pembangunan aplikasi untuk android dan iOS menggunakan satu basis kode (*single codebase*). Pendekatan tersebut dapat mengurangi kompleksitas pengembangan dan pemeliharaan ketika aplikasi ditujukan untuk lebih dari satu platform. Pertimbangan pemilihan Flutter dibandingkan pendekatan *native* maupun *framework cross-platform* lainnya diuraikan lebih lanjut pada subbab 2.1.3, dengan mempertimbangkan konsistensi antarmuka, penggunaan kembali komponen, serta efisiensi pengembangan aplikasi.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, beberapa permasalahan yang dapat diidentifikasi adalah sebagai berikut :

1. Distribusi dan penjualan publikasi ilmiah tekMIRA masih bergantung pada mekanisme konvensional sehingga jangkauan distribusi terbatas dan tidak dapat berlangsung secara berkelanjutan.
2. Pencatatan transaksi yang dilakukan secara konvensional oleh dua unit berbeda menyebabkan ketidakcocokan data dan keterlambatan pelaporan kepada manajemen.
3. Belum tersedia platform digital terpusat yang mengintegrasikan seluruh proses penjualan, pencatatan, dan pelaporan publikasi tekMIRA secara daring.

1.3 Tujuan Tugas Akhir

Tujuan dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Membangun aplikasi toko buku *online* berbasis *mobile* menggunakan Flutter pada studi kasus Balai Besar Pengujian Mineral dan Batubara tekMIRA yang mendukung proses penjualan, pencatatan transaksi, dan pelaporan penjualan secara terpusat.
2. Mengimplementasikan aplikasi *mobile* yang terintegrasi dengan sistem *backend* melalui layanan REST API.

3. Menyusun dokumentasi pembangunan aplikasi secara sistematis sesuai dengan kaidah penulisan ilmiah Tugas Akhir.

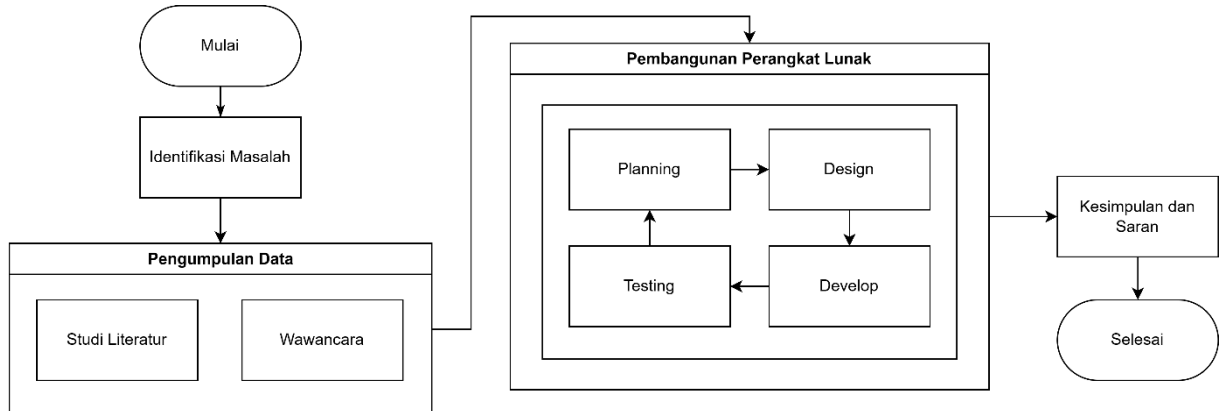
1.4 Lingkup Tugas Akhir

Agar pembahasan Tugas Akhir terarah dan sesuai dengan tujuan yang ditetapkan, maka ruang lingkup pembahasan dibatasi sebagai berikut:

1. Pembangunan aplikasi difokuskan pada sisi aplikasi *mobile* berbasis Android dan tidak mencakup proses *deployment* aplikasi ke lingkungan produksi.
2. Integrasi dengan sistem *backend* dilakukan dengan menggunakan REST API yang telah disediakan, dan pembahasan tidak mencakup pengembangan *backend*.
3. Tidak mencakup integrasi dengan *payment gateway*.
4. Tidak sampai *acceptance testing*.
5. Aplikasi dibangun menggunakan *framework* Flutter versi 3.35.1 dengan bahasa pemrograman Dart.

1.5 Metodologi Pengerjaan Tugas Akhir

Di bawah ini merupakan gambar dan penjelasan mengenai metodologi yang digunakan dalam langkah penyelesaian Tugas Akhir :



Gambar 1. 1 Metodologi Pengerjaan Tugas Akhir

Berdasarkan gambar tersebut, penjelasan setiap tahapan adalah sebagai berikut.

1. Identifikasi Masalah

Pada tahap ini dilakukan pengidentifikasian masalah yang terjadi, serta solusi sementara yang akan diusulkan untuk mengatasi masalah tersebut

2. Pengumpulan Data

Tahap pengumpulan data bertujuan untuk memperoleh informasi yang dibutuhkan sebagai dasar analisis dan pembangunan sistem. Pengumpulan data dilakukan melalui pendekatan sebagai berikut.

a) Studi Literatur

Studi literatur dilakukan dengan mempelajari dan mengkaji referensi ilmiah yang relevan, meliputi konsep *electronic commerce*, pengembangan aplikasi *mobile*, *framework* flutter, arsitektur *client-server*, serta integrasi REST API. Sumber referensi yang digunakan mencakup buku teks, jurnal ilmiah, dan publikasi akademik yang berkaitan dengan topik penelitian

b) Wawancara

Wawancara dilakukan dengan pihak tekMIRA untuk memperoleh data primer mengenai mekanisme penjualan publikasi yang berjalan, kendala operasional yang dihadapi, serta kebutuhan dan ekspektasi terhadap sistem yang akan dibangun. Hasil wawancara menjadi acuan utama dalam pendefinisian kebutuhan fungsional aplikasi.

3. Pembangunan Perangkat Lunak

Pembangunan perangkat lunak dilaksanakan menggunakan pendekatan iteratif yang terdiri atas empat tahapan yang berjalan secara siklus. Apabila pada tahap pengujian ditemukan ketidaksesuaian dengan kebutuhan yang telah ditetapkan, proses dikembalikan ke tahap perencanaan untuk dilakukan penyesuaian pada iterasi berikutnya. Keempat tahapan tersebut adalah sebagai berikut.

a) *Planning*

Tahap *planning* merupakan tahap perencanaan yang dilakukan di awal setiap iterasi. Pada tahap ini didefinisikan kebutuhan fungsional dan non-fungsional sistem berdasarkan hasil pengumpulan data, serta ditetapkan fitur yang akan dikembangkan pada iterasi tersebut.

b) *Design*

Tahap *design* mencakup perancangan arsitektur aplikasi, pemodelan sistem menggunakan diagram UML, serta perancangan antarmuka pengguna (*user interface*) dan alur navigasi aplikasi berdasarkan kebutuhan yang telah didefinisikan pada tahap sebelumnya.

c) *Develop*

Tahap *develop* merupakan tahap implementasi, yaitu pengkodean aplikasi menggunakan *framework* Flutter berdasarkan hasil perancangan yang telah disusun. Implementasi mencakup seluruh fitur pada sisi pengguna maupun admin, serta integrasi dengan sistem *backend* melalui layanan REST API.

d) *Testing*

Tahap *testing* dilakukan untuk memverifikasi bahwa seluruh fungsi aplikasi yang telah diimplementasikan berjalan sesuai dengan kebutuhan yang telah ditetapkan. Pengujian dilakukan secara fungsional terhadap setiap fitur yang dikembangkan dalam setiap iterasi.

4. Kesimpulan dan Saran

Pada tahap ini akan dilakukan penyimpulan dari penelitian yang telah dilakukan terkait dengan masalah yang sudah diidentifikasi, serta sebagai prospek penelitian selanjutnya.

1.6 Sistematika Penulisan Laporan Tugas Akhir

Sistematika penulisan laporan Tugas Akhir disusun untuk memberikan gambaran umum mengenai alur pembahasan dan keterkaitan antar bab dalam laporan ini. Penyusunan sistematika

dilakukan secara terstruktur agar pembahasan dapat dipahami secara runtut, mulai dari pengenalan permasalahan hingga penarikan kesimpulan akhir. Adapun sistematika penulisan laporan Tugas Akhir ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini membahas gambaran umum mengenai tugas akhir yang dikerjakan. Pembahasan meliputi latar belakang masalah, identifikasi masalah, tujuan tugas akhir, lingkup pembahasan, metodologi pengerjaan tugas akhir, serta sistematika penulisan laporan sebagai pengantar terhadap keseluruhan isi laporan.

BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU

Bab ini membahas teori, konsep, dan definisi yang menjadi dasar dalam pengerjaan tugas akhir. Selain itu, bab ini juga mengulas penelitian terdahulu yang relevan, khususnya jurnal ilmiah atau karya penelitian sebelumnya yang memiliki keterkaitan dengan topik pembangunan aplikasi *mobile* dan studi kasus yang digunakan.

BAB 3 SKEMA PENELITIAN

Bab ini menjelaskan rancangan penelitian yang digunakan dalam penyusunan tugas akhir. Pembahasan mencakup alur penyelesaian tugas akhir, analisis permasalahan dan manfaat penelitian, kerangka pemikiran teoritis, serta profil penelitian yang meliputi objek dan batasan penelitian.

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini membahas tahapan awal dalam proses pembangunan aplikasi, yang meliputi analisis kebutuhan sistem dan perancangan sistem. Pada bab ini dijelaskan pendefinisian perangkat lunak, analisis kebutuhan fungsional dan non-fungsional, serta perancangan arsitektur dan spesifikasi sistem sebagai dasar implementasi.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas proses implementasi sistem berupa pengkodean aplikasi berdasarkan hasil analisis dan perancangan yang telah dilakukan. Selain itu, pada bab ini juga dijelaskan proses pengujian sistem untuk memastikan bahwa aplikasi yang dibangun telah memenuhi kebutuhan dan spesifikasi yang ditetapkan.

BAB 6 PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil pelaksanaan tugas akhir serta saran yang dapat dijadikan acuan untuk pengembangan sistem atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI DAN PENELITIAN TERDAHULU

2.1 Teori Pendukung

2.1.1 E-Commerce

Electronic commerce (EC) merujuk pada penggunaan internet dan intranet untuk membeli, menjual, memindahkan, atau memperdagangkan data, barang, maupun jasa [TUR15]. Dalam pengertian yang lebih luas, EC tidak hanya mencakup transaksi jual-beli, tetapi meliputi seluruh aktivitas bisnis yang difasilitasi secara elektronik, termasuk komunikasi, kolaborasi, layanan pelanggan, dan distribusi informasi [TUR15]. Berdasarkan sifat transaksinya, EC diklasifikasikan menjadi *pure EC* apabila seluruh aktivitas pemesanan, pembayaran, pemenuhan pesanan, dan pengiriman dilakukan secara digital, serta *partial EC* apabila paling sedikit satu aktivitas masih bersifat fisik [TUR15]. Berdasarkan jenis hubungan antar pelaku, model yang relevan bagi penelitian ini adalah *Business-to-Consumer* (B2C), yaitu transaksi ritel produk atau jasa dari pelaku bisnis kepada konsumen individual, yang juga dikenal sebagai *e-tailing* [TUR15].

Perkembangan EC dapat ditelusuri sejak awal tahun 1970-an melalui transfer dana elektronik dan *Electronic Data Interchange* (EDI), kemudian mengalami akselerasi pada awal 1990-an seiring komersialisasi *World Wide Web* [TUR15]. Pada tahun 2001 EC mulai merambah ke ranah *mobile commerce* (*m-commerce*), dan pertumbuhan penggunaan *smartphone* serta tablet kemudian menjadi pendorong utama pergeseran EC menuju platform *mobile* [TUR15]. Fase terkini dari evolusi ini adalah *mobile commerce* (*m-commerce*), yang didefinisikan sebagai pelaksanaan EC menggunakan perangkat *mobile* dan jaringan nirkabel yang mencakup transaksi B2C, transfer informasi, serta pembayaran daring [TUR15]. *M-commerce* memiliki lima atribut fundamental yang membedakannya dari EC berbasis *desktop*, yaitu ubikuitas (*ubiquity*), kenyamanan dan kapabilitas, interaktivitas, personalisasi, dan lokalisasi, di mana kelima atribut ini memungkinkan layanan diakses kapan saja, di mana saja, oleh pengguna yang tersebar secara geografis [TUR15].

Sebuah sistem EC daring (*webstore*) terdiri dari komponen-komponen utama yang saling terintegrasi. Komponen tersebut meliputi katalog elektronik (*e-catalog*), keranjang belanja elektronik (*electronic shopping cart*), gerbang pembayaran (*payment gateway*) untuk memproses transaksi secara aman, serta layanan pelanggan (*customer services*) yang mencakup informasi produk dan manajemen hubungan pelanggan [TUR15]. Turban dkk. juga membedakan antara *front end*, yaitu antarmuka tempat pelanggan berinteraksi dengan sistem mulai dari katalog hingga pembayaran, dengan *back end* yang menangani pemenuhan pesanan, manajemen inventaris, dan pemrosesan data [TUR15]. Dalam penelitian ini, aplikasi Flutter yang dibangun berperan sebagai *front end* yang berkomunikasi dengan *backend* melalui REST API, sehingga arsitektur sistem mencerminkan pembagian peran tersebut secara langsung.

2.1.2 Mobile Application

Aplikasi *mobile* (*mobile application*) merupakan perangkat lunak yang dirancang dan dikembangkan khusus untuk dijalankan pada perangkat bergerak seperti *smartphone* dan tablet. Wasserman mendefinisikan *software engineering* untuk aplikasi *mobile* sebagai proses di mana individu atau tim mengorganisasi dan mengelola pembangunan sistem berbasis perangkat lunak, mulai dari konsep hingga satu atau lebih rilis formal, dengan mempertimbangkan karakteristik unik lingkungan *mobile* [WAS10]. Wasserman mengidentifikasi bahwa pengembangan aplikasi *mobile* memiliki perbedaan mendasar dibanding pengembangan perangkat lunak konvensional, antara lain karena adanya penanganan sensor (*sensor handling*) seperti GPS, akselerometer, dan kamera; potensi interaksi dengan aplikasi lain pada perangkat yang sama; keterbatasan sumber daya daya baterai; serta kompleksitas pengujian yang lebih tinggi akibat keragaman perangkat keras dan sistem operasi [WAS10].

Berdasarkan pendekatan pengembangan dan cara eksekusinya, Wasserman mengklasifikasikan aplikasi *mobile* ke dalam tiga kategori utama :

1. Aplikasi *native* adalah aplikasi yang dibangun secara eksklusif untuk satu platform tertentu, misalnya khusus iOS atau khusus Android, menggunakan SDK dan bahasa pemrograman yang ditetapkan oleh masing-masing platform. Aplikasi jenis ini mampu mengoptimalkan seluruh fitur perangkat keras dan antarmuka platform, namun memerlukan pengembangan dan pemeliharaan yang terpisah untuk setiap platform sehingga berimplikasi pada biaya dan waktu yang lebih besar [WAS10].
2. Aplikasi *mobile web* adalah aplikasi yang memiliki klien berbasis perangkat kecil sedangkan eksekusinya dilakukan pada server jarak jauh melalui *browser*, sehingga hanya memerlukan satu basis kode namun bergantung pada konektivitas jaringan [WAS10].
3. Aplikasi *hybrid* dan *cross-platform* merupakan pendekatan yang menggunakan satu atau beberapa lapisan abstraksi (*layer of abstraction*) yang memetakan satu basis kode tunggal (*write once*) menjadi program yang dapat dijalankan secara *native* pada berbagai platform sekaligus [WAS10].

Wasserman mencatat bahwa pada saat penelitiannya, terdapat setidaknya lima platform *mobile* penting yang harus dipertimbangkan, dan dukungan terhadap beberapa platform sekaligus sangat mahal apabila dilakukan dengan pendekatan *native* per platform [WAS10].

Dalam penelitian ini, pendekatan *cross-platform* dipilih sebagai landasan pengembangan menggunakan *framework* Flutter. Pilihan ini selaras dengan pernyataan Wasserman bahwa pendekatan *cross-platform* bertujuan meminimalkan biaya pengembangan dan pemeliharaan sembari mempertahankan kemampuan berjalan di berbagai platform [WAS10]. Wasserman juga menegaskan bahwa desain antarmuka pengguna (*user interface*) memegang peranan yang semakin kritis dalam aplikasi *mobile* karena keterbatasan ruang layar dan perbedaan paradigma interaksi dibanding komputer

desktop, paradigma *mobile* bertumpu pada *widget*, sentuhan (*touch*), dan gerakan fisik alih-alih antarmuka WIMP (*Windows, Icons, Menus, Pointer*) yang lazim pada komputer *desktop* [WAS10]. Hal ini menjadi pertimbangan utama dalam perancangan antarmuka aplikasi toko buku tekMIRA agar fungsionalitas pencarian katalog, proses transaksi, dan pengelolaan pesanan dapat diakses secara intuitif melalui layar *smartphone*.

Dari sisi arsitektur, penelitian ini menerapkan pola *client-server*, di mana aplikasi Flutter pada perangkat *mobile* bertindak sebagai *client* yang mengirim permintaan dan menampilkan antarmuka, sementara *server backend* mengelola data katalog, transaksi, dan pesanan melalui layanan REST API. Wasserman mengingatkan bahwa aplikasi dengan arsitektur semacam ini memerlukan perhatian khusus pada keandalan koneksi jaringan, integritas data saat terjadi pemutusan koneksi, serta mekanisme penanganan pengecualian yang memadai [WAS10].

2.1.3 Framework Flutter



Gambar 2. 1 Flutter

Flutter merupakan sebuah *mobile software development kit* (SDK) yang dibangun dan bersifat *open source*, dikembangkan oleh Google, dengan tujuan utama memberdayakan pengembang untuk membangun aplikasi *mobile* berkualitas tinggi [WIN20]. Windmill mendefinisikan Flutter sebagai sebuah platform yang menyediakan seluruh kebutuhan untuk membangun aplikasi secara lengkap, mencakup *rendering engine*, komponen antarmuka pengguna, *testing framework*, *router*, serta berbagai fitur lainnya, sehingga pengembang dapat berfokus pada permasalahan fungsional aplikasi tanpa harus mengelola infrastruktur dasar secara terpisah [WIN20].

Keunggulan utama Flutter terletak pada kemampuannya untuk menghasilkan aplikasi *cross-platform* dari satu basis kode tunggal (*single codebase*). Berbeda dengan pendekatan berbasis JavaScript yang masih memerlukan *JavaScript bridge* sebagai jembatan antara kode aplikasi dan kode *native* perangkat, Flutter dikompilasi langsung menjadi kode *native* ARM dan dilengkapi dengan *rendering engine* miliknya sendiri, sehingga tidak memerlukan lapisan penerjemah pada saat aplikasi berjalan [WIN20]. Hal ini menjadikan aplikasi Flutter berperforma setara dengan aplikasi *native* yang dikembangkan secara khusus untuk masing-masing platform.

Perbandingan Flutter dengan *framework cross-platform* lain, khususnya React Native yang juga banyak digunakan dalam pengembangan aplikasi *mobile*, menunjukkan sejumlah perbedaan mendasar pada tingkat arsitektur dan konsekuensi teknis yang ditimbulkannya. Gülcüoğlu dkk. mencatat bahwa

React Native dibangun di atas *framework* React berbasis JavaScript yang pada mulanya dikembangkan untuk aplikasi web, sehingga kode JavaScript tersebut memerlukan proses kompilasi tambahan melalui komponen *native* pada masing-masing platform sebelum dapat dieksekusi[GÜL21]. Kondisi tersebut berbeda dengan Flutter yang dikompilasi langsung menjadi kode *native*, tanpa lapisan penerjemah tambahan pada saat aplikasi dijalankan [WIN20]. Hasil pengujian ukuran aplikasi sederhana turut menunjukkan bahwa aplikasi Flutter berukuran 4,7 MB, lebih kecil dibandingkan aplikasi React Native yang berukuran 7 MB untuk fungsionalitas yang setara, sementara React Native memiliki tingkat ketergantungan yang lebih tinggi terhadap pustaka pihak ketiga ketika aplikasi memerlukan akses ke modul *native* seperti Bluetooth, WiFi, atau SMS, dibandingkan Flutter yang menyediakan akses tersebut melalui *widget* bawaan dengan ketergantungan pihak ketiga yang lebih rendah [GÜL21]. Ringkasan perbandingan ketiga pendekatan pengembangan aplikasi *mobile* disajikan pada Tabel 2. 1.

Tabel 2. 1 Perbandingan Pendekatan Pengembangan Aplikasi Mobile

Aspek	Native	React Native	Flutter
Bahasa pemrograman	Kotlin/Java, Swift/Objective-C	JavaScript	Dart
Basis kode	Terpisah per platform	Tunggal, dieksekusi melalui <i>bridge</i> ke komponen <i>native</i>	Tunggal, dikompilasi langsung ke kode <i>native</i>
Ukuran aplikasi (studi kasus)	—	7 MB	4,7 MB
Ketergantungan pustaka pihak ketiga untuk modul <i>native</i>	Rendah	Tinggi	Rendah
Kebutuhan basis kode terpisah per platform	Ya	Tidak	Tidak

Berdasarkan perbandingan tersebut, pemilihan Flutter pada pengembangan aplikasi toko buku *online* tekMIRA mempertimbangkan kebutuhan basis kode tunggal untuk menekan biaya pengembangan, ukuran aplikasi yang lebih ringkas untuk mengakomodasi keragaman spesifikasi perangkat Android pengguna, serta ketergantungan pihak ketiga yang relatif rendah untuk kebutuhan integrasi kamera pada fitur verifikasi mahasiswa dan Firebase Cloud Messaging pada fitur notifikasi. Pendekatan pengembangan *native* terpisah untuk Android dan iOS tidak dipilih karena akan menghasilkan dua basis kode independen yang meningkatkan kebutuhan sumber daya pengembang dan waktu pengerjaan, suatu kondisi yang kurang sesuai dengan keterbatasan anggaran instansi pemerintah sebagaimana tekMIRA. Dengan demikian, Flutter dipandang sebagai pendekatan yang paling seimbang di antara ketiga alternatif tersebut untuk memenuhi kebutuhan fungsional maupun batasan sumber daya penelitian ini.

Flutter menggunakan bahasa pemrograman Dart, yang juga dikembangkan oleh Google. Dart mendukung dua mode kompilasi, yaitu *ahead-of-time* (AOT) dan *just-in-time* (JIT). Kompilasi AOT mengubah kode Dart menjadi kode *native* yang efisien untuk keperluan produksi, sementara kompilasi JIT memungkinkan fitur *hot reload* yang mempercepat siklus pengembangan secara signifikan [WIN20]. Dengan *hot reload*, perubahan kode yang dilakukan pengembang dapat tercermin pada tampilan aplikasi hampir secara instan, tanpa perlu memulai ulang proses kompilasi secara penuh.

Konsep fundamental dalam Flutter adalah bahwa segala sesuatu dalam antarmuka aplikasi direpresentasikan sebagai *widget*. Windmill menyatakan bahwa *widget* dalam Flutter merupakan kelas Dart yang mendeskripsikan tampilan antarmuka, di mana *widget* dapat mendefinisikan tata letak, gaya, animasi, maupun elemen struktural seperti tombol dan bidang teks [WIN20]. Aplikasi Flutter dibangun dengan cara menyusun *widget-widget* tersebut membentuk sebuah hierarki yang disebut *widget tree*, serupa dengan struktur DOM pada pengembangan web.

Flutter mengenal dua jenis *widget* utama, yaitu *StatelessWidget* dan *StatefulWidget*. *StatelessWidget* adalah *widget* yang bersifat statis dan tidak memiliki kondisi internal yang dapat berubah selama masa hidupnya; konfigurasinya sepenuhnya bergantung pada *widget* induknya dan bersifat *immutable* [WIN20]. Sebaliknya, *StatefulWidget* dirancang untuk mengelola kondisi (*state*) internal yang dapat berubah. Setiap *StatefulWidget* memiliki objek *State* yang terasosiasi, di mana objek inilah yang memiliki metode *build* dan mampu memicu proses *render* ulang melalui pemanggilan metode *setState* ketika kondisi internalnya berubah [WIN20].

Flutter versi 3.35.1 digunakan dalam penelitian ini sebagai *framework* utama pembangunan aplikasi toko buku *online* berbasis *mobile* pada studi kasus tekMIRA.

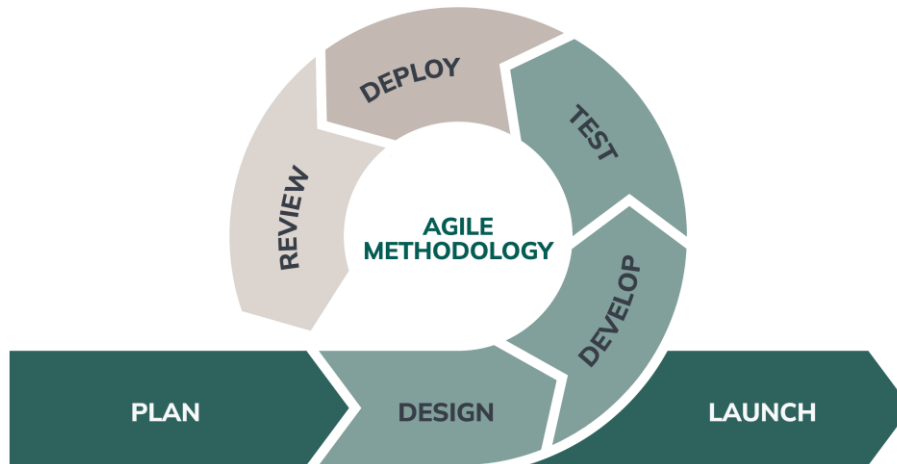
2.1.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) didefinisikan oleh IEEE sebagai penerapan pendekatan yang sistematis, terdisiplin, dan terukur (*systematic, disciplined, quantifiable*) terhadap pengembangan, pengoperasian, dan pemeliharaan perangkat lunak [PRE10]. Pressman menegaskan bahwa rekayasa perangkat lunak merupakan sebuah teknologi berlapis (*layered technology*) yang bertumpu pada komitmen terhadap kualitas, dengan tiga lapisan utama yaitu proses (*process*), metode (*methods*), dan alat bantu (*tools*) yang saling mendukung dalam menghasilkan perangkat lunak yang andal dan memenuhi kebutuhan pengguna [PRE10]. Tanpa disiplin rekayasa perangkat lunak, pengembangan sistem yang kompleks berisiko mengalami keterlambatan, pembengkakan biaya, dan kegagalan dalam memenuhi kebutuhan fungsional maupun kualitas yang diharapkan.

Dalam kerangka tersebut, proses rekayasa perangkat lunak berfungsi sebagai kerangka kerja (*framework*) yang menghubungkan seluruh lapisan teknologi dan mengarahkan aktivitas pengembangan secara terstruktur [PRE10]. Pressman mengusulkan model proses generik yang terdiri

atas lima aktivitas kerangka (*framework activities*), yaitu komunikasi (*communication*), perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), dan penerapan (*deployment*) [PRE10]. Kelima aktivitas ini berlaku untuk semua jenis dan skala proyek perangkat lunak, dan dapat diorganisasikan dalam berbagai pola alur proses, seperti alur linier, iteratif, evolusioner, maupun paralel, sehingga organisasi dapat memilih dan mengadaptasi model yang paling sesuai dengan kebutuhan proyeknya [PRE10].

2.1.5 Pengembangan Perangkat Lunak *Agile*



Gambar 2. 2 Agile Methodology

Seiring berkembangnya kebutuhan yang berubah cepat dan dinamis, pendekatan model proses preskriptif (*prescriptive process models*) yang bersifat kaku dinilai kurang memadai untuk menghadapi perubahan kebutuhan selama pengembangan berlangsung [PRE10]. Sebagai respons terhadap keterbatasan tersebut, muncul paradigma *agile software development* yang menekankan fleksibilitas, kolaborasi, dan pengiriman perangkat lunak secara inkremental [PRE10]. Pressman menjelaskan bahwa pendekatan *agile* berangkat dari manifesto pengembangan *agile* yang menetapkan empat nilai utama, yaitu fokus pada individu dan interaksi di atas proses dan alat, perangkat lunak yang berfungsi di atas dokumentasi yang komprehensif, kolaborasi dengan pelanggan di atas negosiasi kontrak, serta kemampuan merespons perubahan di atas kepatuhan terhadap rencana [PRE10].

Nilai-nilai tersebut diwujudkan melalui dua belas prinsip *agility* yang ditetapkan oleh Agile Alliance, di antaranya penyerahan perangkat lunak yang berfungsi secara berkala dalam rentang waktu singkat, penerimaan terhadap perubahan kebutuhan bahkan pada tahap akhir pengembangan, serta refleksi dan penyesuaian cara kerja tim secara berkala [PRE10]. Pressman menyatakan bahwa biaya perubahan dalam proses *agile* dapat ditekan secara signifikan karena perangkat lunak dirilis dalam bentuk inkremen, sehingga perubahan dapat dikendalikan dalam lingkup inkremen tersebut tanpa harus mengubah keseluruhan sistem [PRE10]. Selain itu, proses *agile* yang efektif harus mampu menyesuaikan tugas-tugas pengembangan, mengelola alur kerja secara efisien, dan tetap menjaga

keseimbangan antara kecepatan penyerahan perangkat lunak dengan kualitas produk yang dihasilkan [PRE10].

Dalam penelitian ini, metodologi pengembangan yang digunakan mengadopsi pendekatan iteratif yang mengacu pada siklus *agile methodology*. Siklus *agile* secara umum mencakup tahapan *plan*, *develop*, *test*, *deploy*, *review*, hingga *launch*, sebagaimana diilustrasikan pada Gambar 2.1. Namun, sesuai dengan lingkup penelitian yang telah ditetapkan, hanya empat tahapan yang diterapkan, yaitu *plan*, *design*, *develop*, dan *test*, di mana tahapan *develop* dan *test* dijalankan secara iteratif hingga seluruh fitur yang direncanakan berhasil dibangun dan diverifikasi. Tahapan *deploy*, *review*, dan *launch* tidak termasuk dalam lingkup penelitian ini, sehingga siklus pengembangan dibatasi pada pembangunan dan pengujian aplikasi.

2.1.6 Firebase



Gambar 2. 3 Firebase

Firebase merupakan platform pengembangan aplikasi berbasis *cloud* yang dikembangkan oleh Google, dirancang untuk menyediakan infrastruktur dan layanan yang dibutuhkan pengembang dalam membangun aplikasi berkualitas tinggi [MOR17]. Moroney mendefinisikan Firebase sebagai sekumpulan layanan yang berjalan di atas infrastruktur Google, yang mencakup tiga pilar utama: *Develop*, *Grow*, dan *Earn*, di mana setiap teknologi di dalamnya saling terhubung melalui analitik terpadu [MOR17]. Firebase bukan pengganti API yang ada, melainkan sebuah peningkatan (*enhancement*) yang menyediakan layanan umum seperti basis data, *autentikasi*, pesan, dan lainnya, sehingga pengembang dapat berfokus pada keunikan aplikasi yang dibangun [MOR17].

2.1.7 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) merupakan solusi pengiriman pesan lintas platform (*cross-platform messaging solution*) yang dikembangkan Google, memungkinkan pengiriman pesan dan notifikasi secara andal dan gratis ke aplikasi klien yang berjalan pada platform Android, iOS, maupun web [FCM26]]. Moroney menyatakan bahwa lebih dari 98% perangkat yang terhubung menerima pesan FCM dalam waktu kurang dari 500 milidetik, menjadikannya komponen infrastruktur yang kritis dalam aplikasi yang mengutamakan komunikasi *real-time* [MOR17].

FCM mendukung dua jenis pesan utama [FCM26]:

1. *Notification messages*, yaitu pesan yang ditampilkan langsung kepada pengguna di area notifikasi perangkat dan dikelola secara otomatis oleh FCM SDK.

2. *Data messages*, yaitu pesan yang membawa muatan data (*payload*) hingga 4.096 byte langsung ke aplikasi klien untuk diproses oleh logika aplikasi.

Secara arsitektural, FCM terdiri dari dua komponen utama yang bekerja secara sinergis [FCM26]. Komponen pertama adalah *trusted environment*, yaitu server aplikasi yang membangun, menargetkan, dan mengirimkan pesan menggunakan Firebase Admin SDK atau protokol server FCM. Komponen kedua adalah aplikasi klien pada perangkat yang menerima pesan melalui Firebase Cloud Messaging SDK [FCM26].

Dalam penelitian ini, FCM diintegrasikan ke dalam aplikasi Flutter untuk mendukung alur notifikasi sebagai berikut:

1. Notifikasi kepada admin ketika *customer* melakukan pembelian dan mengirimkan bukti pembayaran.
2. Notifikasi kepada admin ketika *customer* mengajukan permintaan verifikasi status pelajar.
3. Notifikasi kepada *customer* ketika admin memverifikasi pembayaran, memverifikasi status pelajar, dan memperbarui status pesanan beserta nomor resi pengiriman.

2.2 Penelitian Terdahulu

Tabel 2. 2 Penelitian Terdahulu

No.	Judul Penelitian	Peneliti	Hasil Penelitian
1.	Pengembangan Aplikasi Toko Buku berbasis Web (Studi Kasus: Toko Buku Online Yukmoco)	Bregaster, Denny Sagita Rusdianto, Edy Santoso [BRE22]	Sistem <i>e-commerce</i> berhasil mengakomodasi pembelian dari lebih dari satu distributor dalam satu transaksi; pengujian <i>usability</i> SUS menghasilkan nilai 79 (<i>acceptable</i>).
2.	Sistem Informasi Perencanaan Aplikasi Penjualan Buku Online Berbasis Android	Rizki Nanda Winata Prasetya, Muhammad Sukron, Rizki Hidayat Pane, Muhammad Hafiz Al Fazri, Muhammad Alda [PRE25]	Aplikasi berhasil menyediakan fitur katalog buku berdasarkan kategori, buku terlaris, dan terbaru sehingga konsumen mudah menemukan informasi buku; fitur transaksi penjualan <i>online</i> membantu proses penjualan kepada konsumen.
3.	Perancangan Aplikasi Penjualan Buku Berbasis Android pada CV Kamalia Sinar Sejahtera	Tuti Hartati, L.M. Azwar Al Fayyadh [HAR25]	Aplikasi mengotomatiskan pengelolaan katalog, pencatatan transaksi, dan pembuatan faktur; rata-rata skor kepuasan pengguna (Likert) 4,47 dari skala 5
4.	Perancangan Sistem Informasi Manajemen Stok dan Penjualan Buku Berbasis Android Pada Musi Bookstore	Adelano Nur Abadi, Farida Ardiani [ABA24]	Aplikasi meningkatkan akurasi dan efisiensi pencatatan transaksi serta pengelolaan stok secara <i>real-time</i> melalui fitur <i>barcode scanning</i> Google ML Kit.
5.	Design and Build a Mobile Bookstore Application in Order to Attract Consumers' Reading Interest Using Prototyping Method	Liyunjiraa, I Putu Dody Lesmana [LIY22]	Aplikasi berhasil menarik minat baca pengguna; skor SUS meningkat dari 80,5 (<i>excellent</i>) ke 90,5 (<i>best imaginable</i>); rata-rata respons pengguna 84% kategori sangat kuat

BAB 3

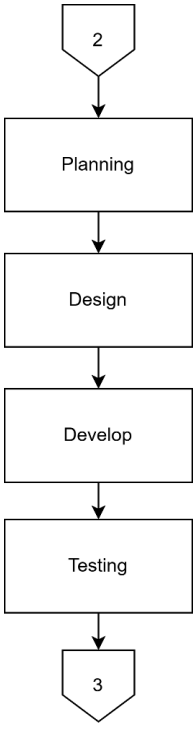
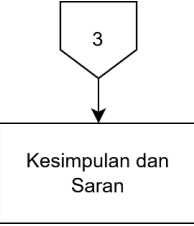
SKEMA PENELITIAN

3.1 Rancangan Penelitian

Rancangan penelitian pada tugas akhir ini menggambarkan keseluruhan alur pelaksanaan pembangunan aplikasi toko buku *online* berbasis *mobile* untuk tekMIRA secara sistematis. Rincian setiap tahapan beserta hasil, kontribusi, langkah penelitian, dan literatur yang digunakan disajikan pada tabel 3.1.

Tabel 3. 1 Alur Penyelesaian Tugas Akhir

Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 1 : Identifikasi Masalah Hasil : Mengetahui permasalahan yang dialami oleh tekMIRA dalam proses distribusi dan penjualan publikasi teknis untuk diselesaikan Kontribusi : Berguna untuk pengumpulan data dan memudahkan dalam penentuan masalah dan lingkup proses penyelesaian masalah	<pre> graph TD A[Menentukan topik Tugas Akhir] --> B[Melakukan wawancara dengan pihak tekMIRA mengenai permasalahan yang dialami] B --> C[Mengetahui permasalahan yang dialami tekMIRA] C --> D[Memahami proses bisnis penjualan publikasi dan kebutuhan pengguna] D --> E{{1}} </pre>	1. Wawancara dengan pihak tekMIRA
Tahap 2 : Pengumpulan Data Hasil : Memperoleh informasi dasar yang dibutuhkan dalam pembangunan perangkat lunak sebagai bentuk penyelesaian masalah. Kontribusi : Berguna untuk pengumpulan data dan memudahkan dalam penentuan masalah dan lingkup proses penyelesaian masalah	<pre> graph TD A{{1}} --> B[Melakukan wawancara dengan pihak tekMIRA] B --> C[Mengumpulkan referensi dan teori yang relevan mengenai mobile e-commerce] C --> D{{2}} </pre>	1. Wawancara dengan pihak tekMIRA 2. <i>E-Commerce</i> [TUR15] 3. <i>Mobile Application</i> [WAS10] 4. <i>Framework Flutter</i> [WIN20] 5. <i>Rekayasa Perangkat Lunak</i> [PRE10]

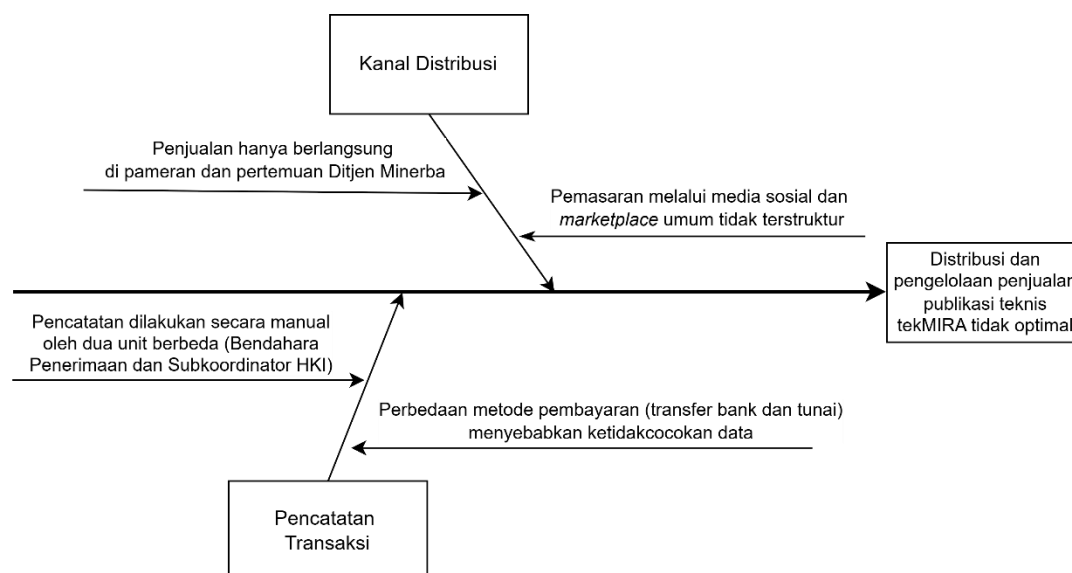
Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 3 : Pembangunan Perangkat Lunak Hasil : Aplikasi toko buku <i>online</i> berbasis <i>mobile</i> dibangun menggunakan <i>framework</i> Flutter dengan metode agile iteratif, yang dimulai dari tahap perancangan (<i>planning</i>), desain (<i>design</i>), pengembangan (<i>develop</i>), dan pengujian (<i>testing</i>). Kontribusi : Menghasilkan artefak utama penelitian berupa aplikasi <i>mobile</i> yang fungsional dan telah diverifikasi melalui pengujian, sehingga tujuan pembangunan sistem dapat terpenuhi secara terukur.	 <pre> graph TD A{{2}} --> B[Planning] B --> C[Design] C --> D[Develop] D --> E[Testing] E --> F{{3}} </pre>	1. Metode Agile [PRE10]. 2. <i>Framework</i> Flutter [WIN20]. 3. <i>Mobile Application</i> [WAS10]. 4. <i>E-Commerce</i> [TUR15].
Tahap 4 : Kesimpulan dan Saran Hasil : Kesimpulan dan saran berdasarkan hasil pembangunan dan pengujian aplikasi toko buku <i>online</i> berbasis <i>mobile</i> tekMIRA yang telah selesai. Kontribusi : Merefleksikan ketercapaian tujuan penelitian dan memberikan rekomendasi pengembangan lanjutan bagi penelitian selanjutnya.	 <pre> graph TD A{{3}} --> B[Kesimpulan dan Saran] </pre>	1. Software Engineering [PRE10]

3.2 Perumusan Masalah

Perumusan masalah bertujuan untuk mengidentifikasi isu utama yang akan dijelajahi dalam penelitian tugas akhir ini, yaitu belum optimalnya distribusi dan pengelolaan penjualan publikasi teknis ilmiah milik Balai Besar Pengujian Mineral dan Batubara tekMIRA.

3.2.1 Analisis Sebab Akibat

Tahap analisis pada bagian ini memanfaatkan diagram sebab-akibat, yang umumnya dikenal sebagai fishbone diagram. Fishbone diagram digunakan sebagai alat untuk mengidentifikasi dan merangkum penyebab yang mungkin muncul dari suatu masalah dalam format terstruktur. Berikut ini adalah representasi fishbone diagram dari analisis masalah yang tertera pada Gambar 3.1.



Gambar 3. 1 Kerangka Permasalahan

Berdasarkan diagram tersebut, teridentifikasi dua aspek permasalahan utama, yaitu kanal distribusi publikasi tekMira yang masih terbatas dan proses pencatatan transaksi yang belum terintegrasi. Rincian penyebab dan akibat dari masing-masing aspek disajikan pada Tabel 3.2.

Tabel 3. 2 Analisis Sebab Akibat

Aspek Permasalahan	Penyebab	Akibat
Kanal Distribusi	Penjualan hanya berlangsung di pameran dan pertemuan Ditjen Minerba	Penjualan hanya berlangsung pada waktu dan lokasi tertentu, sehingga calon pembeli di luar kegiatan tersebut sulit mengakses publikasi.
	Pemasaran melalui media sosial dan <i>marketplace</i> umum belum terstruktur.	Informasi katalog, pemesanan, pencatatan transaksi, pembaruan stok, dan pelaporan belum terhubung dalam satu alur penjualan.
Pencatatan Transaksi	Pencatatan dilakukan secara manual oleh dua unit berbeda (Bendahara Penerimaan dan Subkoordinator HKI).	Data penjualan berpotensi tidak sesuai antara unit yang terlibat.
	Perbedaan metode pembayaran antara transfer bank dan pembayaran tunai di lokasi pameran.	Pembaruan informasi penjualan dan pelaporan kepada manajemen menjadi lebih lambat.

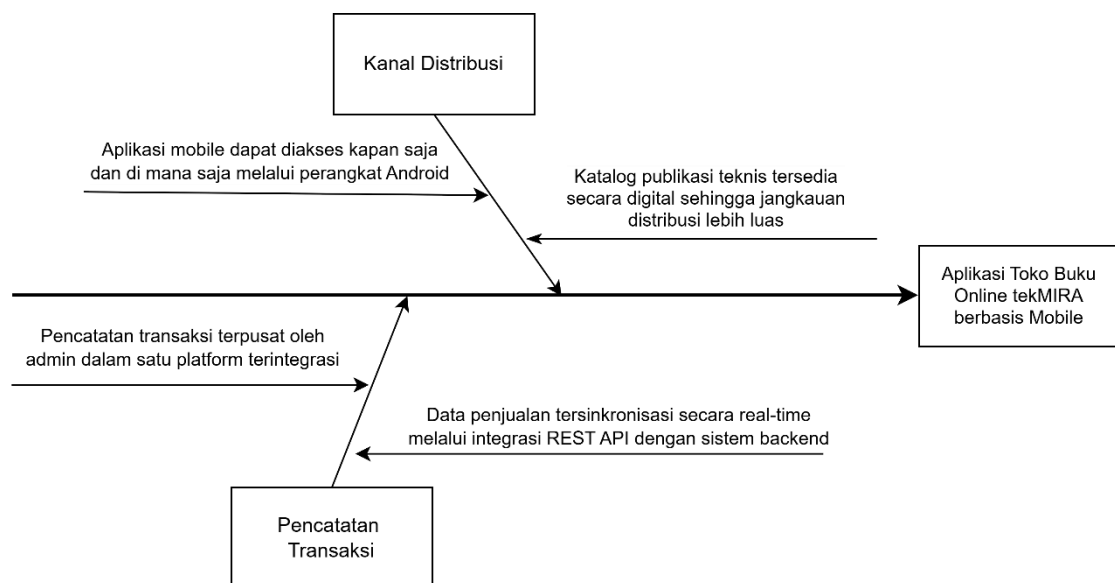
3.2.2 Solusi Masalah

Berdasarkan akar masalah pada subbab sebelumnya, disampaikan solusi yang mungkin dari setiap penyebab tersebut.

Tabel 3. 3 Solusi Masalah

No.	Faktor Penyebab	Solusi
1.	Kanal Distribusi : - Penjualan hanya dapat dilakukan saat pameran dan pertemuan Ditjen Minerba - Pemasaran melalui media sosial dan <i>marketplace</i> umum tidak terstruktur dan tidak berkelanjutan	Membangun aplikasi toko buku <i>online</i> berbasis <i>mobile</i> menggunakan Flutter sebagai <i>framework</i> , sehingga publikasi teknis tekMira dapat diakses dan dibeli kapan saja dan di mana saja melalui perangkat <i>mobile</i> .
2.	Pencatatan Transaksi : - Pencatatan manual oleh dua unit berbeda (Bendahara Penerimaan dan Subkoordinator HKI) - Perbedaan metode pembayaran menyebabkan ketidakcocokan data dan keterlambatan pelaporan	Menyediakan fitur manajemen transaksi terpusat yang memungkinkan admin mencatat seluruh transaksi, baik melalui transfer maupun pembayaran tunai, dalam satu sistem sehingga data penjualan lebih terstruktur dan mudah dipantau

Berdasarkan solusi yang dirumuskan pada Tabel 3.2, gambaran kerangka solusi masalah direpresentasikan menggunakan *fishbone diagram* pada Gambar 3.2 berikut.



Gambar 3. 2 Kerangka Solusi Masalah

Tabel 3. 4 Penjelasan Kerangka Solusi Masalah

No.	Komponen Utama	Detail Komponen	Penjelasan
1.	Kanal Distribusi	- Aplikasi <i>mobile</i> dapat diakses kapan saja dan di mana saja melalui perangkat Android - Katalog publikasi teknis tersedia secara digital dalam bentuk <i>e-catalog</i>	Pembangunan aplikasi toko buku <i>online</i> berbasis <i>mobile</i> menggunakan Flutter memungkinkan publikasi teknis tekMIRA dijangkau oleh pemangku kepentingan industri mineral dan batubara di seluruh Indonesia tanpa batasan waktu dan lokasi. Katalog digital menggantikan ketergantungan pada kehadiran fisik di pameran, sehingga distribusi dapat berlangsung secara berkelanjutan.

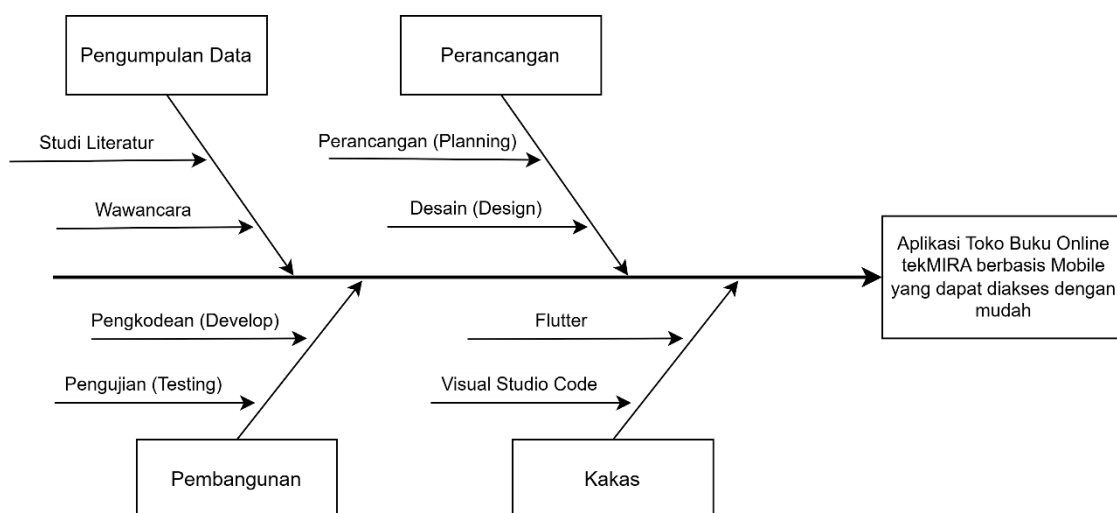
2.	Pencatatan	1. Pencatatan transaksi terpusat oleh admin dalam satu platform 2. Data penjualan tersinkronisasi secara <i>real-time</i> melalui integrasi REST API.	Fitur manajemen transaksi pada aplikasi memungkinkan admin mencatat seluruh penjualan, baik yang terjadi secara daring maupun langsung (on-site) dalam satu sistem yang terintegrasi dengan <i>backend</i> melalui REST API. Data yang tersinkronisasi secara <i>real-time</i> mengeliminasi ketidakcocokan data antara Bendahara Penerimaan dan Subkoordinator HKI, sekaligus mempercepat pelaporan penjualan kepada manajemen.
----	------------	--	--

3.3 Kerangka Berpikir Teoritis

3.3.1 Gambaran Produk Tugas Akhir

Kerangka pemikiran teoritis disusun dengan tujuan mengidentifikasi prinsip-prinsip teoritis yang menjadi dasar dalam menjelaskan fenomena yang sangat teliti, serta untuk mengklarifikasi hubungan antar konsep yang relevan dalam konteks penelitian ini.

Berikut ini adalah kerangka pemikiran teoritis dari penelitian tugas akhir yang direpresentasikan menggunakan *fishbone diagram* pada Gambar 3.3.



Gambar 3. 3 Kerangka Pemikiran Teoritis

Tabel 3. 5 Kerangka Pemikiran Teoritis

No.	Komponen Utama	Detail Komponen	Penjelasan
1.	Pengumpulan Data	1. Studi Literatur 2. Wawancara	Pengumpulan data adalah proses mengumpulkan informasi atau fakta dari berbagai sumber yang digunakan untuk menganalisis kebutuhan, mendukung pengambilan keputusan, dan sebagai dasar dalam menyelesaikan masalah distribusi dan penjualan publikasi teknis tekMIRA.
2.	Perancangan	1. Perancangan (<i>Planning</i>) 2. Desain (<i>Design</i>)	Perancangan merupakan tahap mendefinisikan kebutuhan fungsional dan non-fungsional sistem serta merancang arsitektur aplikasi, diagral UML, dan antarmuka pengguna (<i>user interface</i>) sebagai <i>blueprint</i> teknis sebelum pengkodean dilakukan.

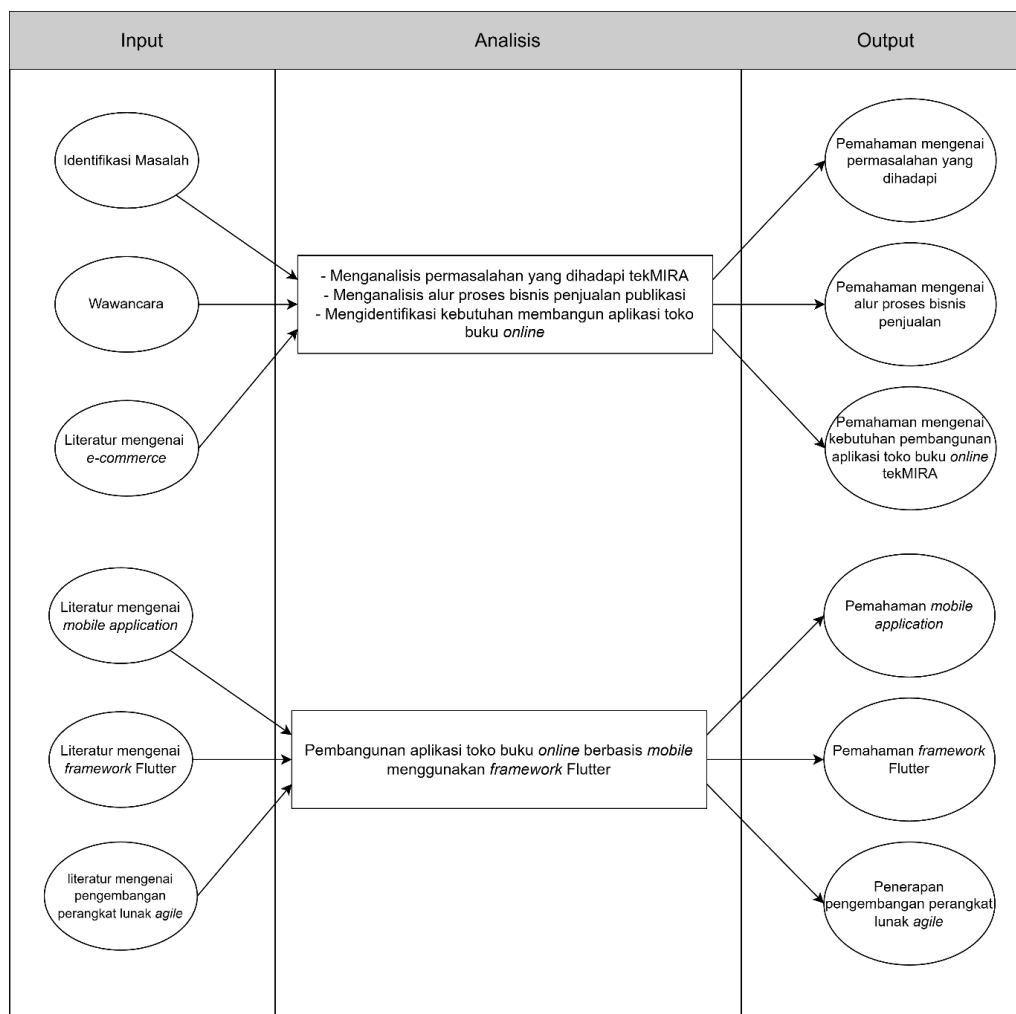
3.	Pembangunan	1. Pengkodean (<i>Develop</i>) 2. Pengujian (<i>Testing</i>)	Pembangunan adalah proses implementasi perangkat lunak berdasarkan hasil perancangan yang telah disusun, yang meliputi pengkodean aplikasi menggunakan <i>framework</i> Flutter serta pengujian fungsional terhadap setiap fitur yang dikembangkan pada setiap iterasi.
4.	Kakas	1. Flutter 2. Visual Studio Code	Kakas merupakan perangkat lunak, aplikasi, atau alat yang digunakan dalam proses pembangunan perangkat lunak. Flutter digunakan sebagai <i>framework</i> utama pengembangan aplikasi <i>cross-platform</i> , sedangkan Visual Studio Code digunakan sebagai <i>code editor</i> .

3.3.2 Skema Analisis Teori

Skema analisis menjelaskan mengenai analisis yang dilakukan selama penelitian tugas akhir. Skema analisis ini memperlihatkan input yang diperlukan untuk kebutuhan analisis dan output dari hasil analisis yang dilakukan. Terdapat tiga komponen utama yang terdapat pada skema analisis yaitu sebagai berikut:

1. *Input*, merupakan komponen yang menjelaskan mengenai masukan yang digunakan sebagai acuan atau referensi untuk tahap analisis. *Input* bisa berupa data dan informasi yang dapat dijadikan dasar analisis.
2. Analisis, merupakan komponen yang menjelaskan mengenai tahapan analisis yang dilakukan dalam penelitian tugas akhir. Tujuannya untuk mengetahui keadaan sebenarnya dalam pengerjaan tugas akhir terkait dengan objek penelitian.
3. *Output*, merupakan komponen yang menjelaskan mengenai keluaran yang dihasilkan dari tahap analisis yang telah dilakukan.

Berikut merupakan skema analisis teori dari penelitian tugas akhir ini yang digambarkan pada Gambar 3.4.



Gambar 3. 4 Skema Analisis Teori

Tabel 3. 6 Detail Skema Analisis Teori

No.	Input	Langkah Analisis	Output	Penjelasan
1.	- Identifikasi Masalah - Wawancara - Literatur mengenai e-commerce [TUR15]	- Menganalisis permasalahan yang dihadapi tekMIRA - Menganalisis alur proses bisnis penjualan publikasi - Mengidentifikasi kebutuhan membangun aplikasi toko buku online	- Pemahaman mengenai permasalahan yang dihadapi - Pemahaman mengenai alur proses bisnis - Pemahaman mengenai kebutuhan membangun aplikasi	Pada tahap ini menjelaskan mengenai identifikasi masalah, literatur yang digunakan, serta proses bisnis penjualan publikasi teknis tekMIRA yang berjalan.
2.	- Literatur mengenai mobile application. - Literatur mengenai framework Flutter - Literatur mengenai pengembangan perangkat lunak agile [PRE10]	- Pembangunan aplikasi toko buku online berbasis mobile menggunakan framework Flutter dengan pendekatan agile iteratif	- Pemahaman mobile application - Pemahaman framework Flutter - Penerapan pengembangan perangkat lunak agile	Pada tahap ini pembangunan aplikasi toko buku online tekMIRA dilakukan menggunakan framework Flutter dengan menerapkan pendekatan pengembangan perangkat lunak agile yang iteratif.

3.4 Profile Penelitian



Gambar 3. 5 Profile Penelitian

Balai Besar Pengujian Mineral dan Batubara tekMIRA merupakan Unit Pelaksana Teknis (UPT) di lingkungan Kementerian Energi dan Sumber Daya Mineral yang melaksanakan tugas teknis operasional dan penunjang di bidang mineral dan batu bara. Beralamat di Jalan Jenderal Sudirman No. 623, Kota Bandung, tekMIRA melaksanakan kegiatan pengujian, penelitian, pengembangan teknologi, serta penyebarluasan hasil penelitian melalui publikasi, termasuk penerbitan buku hasil riset di bidang energi dan sumber daya mineral.

TekMIRA memiliki sejarah yang berawal pada tahun 1956 dengan nama Balai Penyelidikan Mineral di bawah Pusat Djawatan Geologi, Kementerian Perekonomian. Seiring perkembangan fungsi dan tanggung jawab kelembagaan, institusi ini mengalami beberapa perubahan struktur organisasi dan nomenklatur hingga menjadi Balai Besar Pengujian Mineral dan Batubara tekMIRA. Dalam penelitian ini, aktivitas publikasi menjadi fokus karena aplikasi yang dibangun ditujukan untuk mendukung distribusi, pencatatan transaksi, dan pemantauan penjualan publikasi teknis tekMIRA.

Unit kerja yang berkaitan dengan penelitian adalah bagian yang menangani urusan hukum, kerja sama, dan informasi, khususnya dalam pengelolaan informasi dan layanan digital. Kebutuhan tersebut berkaitan dengan pembangunan Aplikasi Toko Buku Online tekMIRA berbasis *mobile* untuk mendukung proses penjualan publikasi secara terpusat. Oleh karena itu, Balai Besar Pengujian Mineral dan Batubara tekMIRA dipilih sebagai objek penelitian karena memiliki kebutuhan terhadap sistem digital yang dapat mendukung distribusi dan pencatatan penjualan publikasi teknis.

Visi

“Terwujudnya BLU tekMira yang Berdaya Saing Tinggi.”

Misi

1. Meningkatkan kemampuan litbang di bidang pertambangan mineral dan batu bara.
2. Meningkatkan produktivitas layanan di bidang pertambangan mineral dan batu bara.
3. Mengembangkan platform bisnis digital berbasis pertambangan mineral dan batu bara.
4. Melaksanakan pengelolaan sumber daya sesuai dengan kaidah kelembagaan yang baik (*good governance*)

BAB 4

ANALISIS DAN PERANCANGAN

4.1 Analisis Kebutuhan

4.1.1 Analisis *Stakeholder*

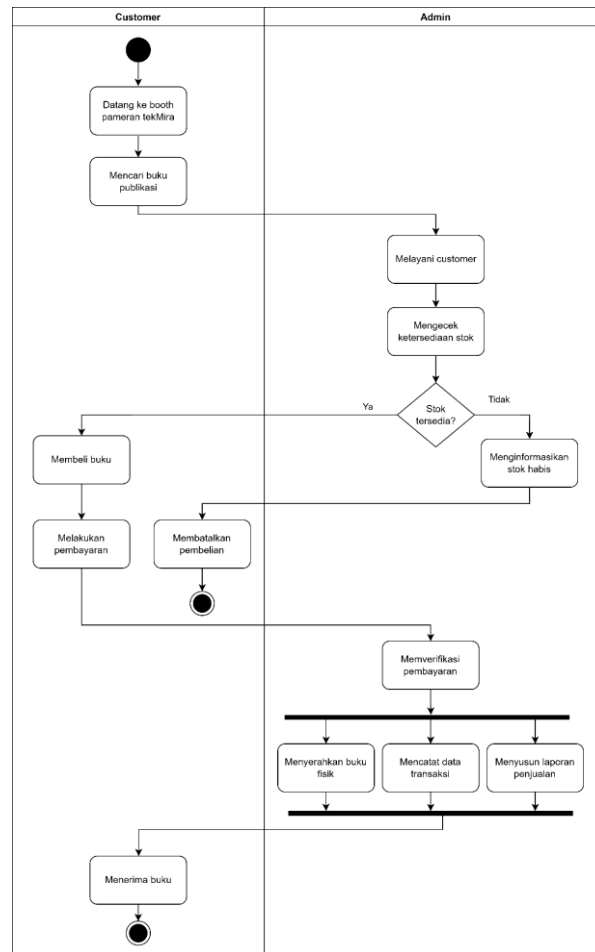
Analisis *stakeholder* dilakukan untuk mengidentifikasi pihak-pihak yang terlibat dan berkepentingan terhadap sistem yang akan dibangun. Pemahaman terhadap peran dan kebutuhan setiap *stakeholder* menjadi landasan dalam mendefinisikan kebutuhan fungsional aplikasi toko buku *online* tekMIRA. Berdasarkan hasil wawancara terhadap proses bisnis yang berjalan di tekMIRA, terdapat dua kategori *stakeholder* utama yang terlibat dalam sistem ini, yaitu pengguna aplikasi dan pengelola sistem.

Tabel 4. 1 Analisis Stakeholder

No.	Stakeholder	Deskripsi
1.	Customer	Pihak eksternal yang menjadi target pengguna aplikasi toko buku <i>online</i> tekMIRA. <i>Customer</i> terdiri dari akademisi, peneliti, pemangku kepentingan industri mineral dan batubara, serta masyarakat umum yang membutuhkan publikasi teknis dari tekMIRA.
2.	Admin	Pihak internal tekMIRA yang bertanggung jawab terhadap pengelolaan operasional sistem. Admin memiliki kepentingan dalam pengelolaan data publikasi, transaksi penjualan, verifikasi mahasiswa, serta pemantauan laporan penjualan agar proses bisnis dapat berjalan lebih terstruktur dan terintegrasi.

4.1.2 Analisis Proses Bisnis Berjalan

Proses bisnis penjualan publikasi yang saat ini berjalan di Balai Besar Pengujian Mineral dan Batubara tekMIRA ditunjukkan pada Gambar 4.1. Proses tersebut menggambarkan alur penjualan publikasi mulai dari pelanggan mencari buku hingga transaksi penjualan dicatat dan dilaporkan.



Gambar 4. 1 Diagram Aktivitas

Berdasarkan Gambar 4.1, penjualan masih dilakukan secara langsung melalui kegiatan pameran, sedangkan pencatatan transaksi dan penyusunan laporan dilakukan secara konvensional menggunakan buku catatan. Kondisi tersebut menyebabkan pengelolaan data penjualan belum terintegrasi dan berpotensi menimbulkan keterlambatan pembaruan informasi. Selain itu, ketergantungan pada interaksi langsung membatasi jangkauan distribusi publikasi kepada calon pembeli dari berbagai wilayah.

Berdasarkan kondisi tersebut, diperlukan aplikasi yang menyediakan kanal penjualan daring, mengelola transaksi secara terpusat, serta mendukung penyajian informasi penjualan secara lebih cepat dan terintegrasi.

4.1.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mendefinisikan kebutuhan fungsional sistem secara terstruktur berdasarkan kebutuhan setiap *stakeholder* yang berinteraksi dengan aplikasi. Berdasarkan hasil analisis *stakeholder* dan proses bisnis yang berjalan, kebutuhan fungsional aplikasi toko buku tekMIRA dapat dirumuskan sebagaimana ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Analisis Kebutuhan Fungsional

Kode (FR)	User Requirement	Kode (SR)	Software Requirement
FR-01	User ingin dapat mendaftarkan akun baru.	SR-01	Perangkat lunak harus dapat memvalidasi data registrasi dan membuat akun pengguna baru.
FR-02	User ingin dapat melihat katalog dan informasi lengkap buku.	SR-02	Perangkat lunak harus dapat menampilkan daftar buku serta informasi lengkap buku yang dipilih, meliputi judul, deskripsi, harga, stok, gambar, dan informasi terkait lainnya.
FR-03	User ingin dapat mengelola keranjang belanja.	SR-03	Perangkat lunak harus dapat menambah, mengubah jumlah, dan menghapus item pada keranjang serta menghitung total harga belanja.
FR-04	User ingin dapat mengelola alamat pengiriman.	SR-04	Perangkat lunak harus dapat menambah, mengubah, menghapus, dan menetapkan alamat utama pengguna.
FR-05	User ingin dapat melakukan pemesanan buku.	SR-05	Perangkat lunak harus dapat memproses pemesanan berdasarkan keranjang, alamat, dan pilihan pengiriman, menghitung ongkos kirim, membuat data pesanan, serta melakukan pembayaran.
FR-06	User ingin dapat melihat riwayat dan detail pesanan.	SR-06	Perangkat lunak harus dapat menampilkan daftar, detail, dan status pesanan pengguna, serta mengirimkan notifikasi ketika terjadi perubahan status pesanan atau pembayaran.
FR-07	Admin ingin dapat melihat ringkasan data penjualan pada <i>dashboard</i> .	SR-07	Perangkat lunak harus dapat menampilkan statistik penjualan, grafik penjualan bulanan, dan daftar buku terlaris pada <i>dashboard</i> admin.
FR-08	Admin ingin dapat mengelola pesanan dan menerima informasi pesanan yang memerlukan tindakan.	SR-08	Perangkat lunak harus dapat menampilkan daftar pesanan, mengonfirmasi pembayaran, memasukkan nomor resi, serta mencatat transaksi penjualan tunai, serta mengirimkan notifikasi kepada admin ketika terdapat pesanan masuk yang memerlukan konfirmasi pembayaran.
FR-09	Admin ingin dapat mengelola data buku.	SR-09	Perangkat lunak harus dapat menambah, mengubah, dan menghapus data buku beserta gambar buku pada katalog.

4.1.4 Analisis Pengguna

Analisis pengguna dilakukan untuk mengidentifikasi aktor yang berinteraksi secara langsung dengan aplikasi toko buku *online* tekMIRA beserta peran dan hak aksesnya dalam sistem. Berdasarkan hasil analisis kebutuhan sistem, terdapat dua jenis aktor utama yang terlibat dalam penggunaan aplikasi. Analisis pengguna tersebut dirangkum pada Tabel 4.3.

Tabel 4. 3 Analisis Pengguna

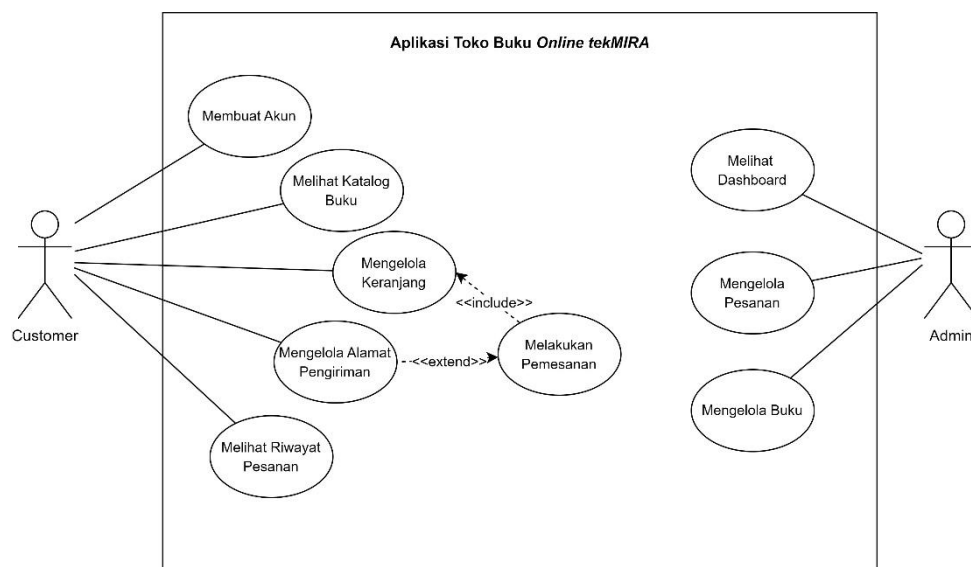
Kode Aktor	Nama Aktor	Deskripsi
A01	Customer	Pengguna aplikasi yang memiliki hak akses untuk melihat katalog buku, melakukan pembelian, mengelola keranjang dan <i>wishlist</i> , melakukan pembayaran, melihat status pesanan, serta memberikan ulasan terhadap buku.
A02	Admin	Pengguna yang memiliki hak akses untuk mengelola data buku, pesanan, diskon, ulasan, verifikasi mahasiswa, transaksi penjualan tunai, serta memantau laporan penjualan melalui <i>dashboard</i> admin.

4.2 Pemodelan berbasis Skenario

Pemodelan berbasis skenario dilakukan untuk menggambarkan interaksi antara aktor dengan sistem berdasarkan kebutuhan fungsional yang telah didefinisikan. Pemodelan ini disajikan dalam bentuk *Use case diagram* dan skenario *Use case* untuk memperjelas fungsi-fungsi utama yang disediakan oleh aplikasi toko buku *online* tekMIRA.

4.2.1 Use case Diagram

Use case diagram digunakan untuk menggambarkan hubungan antara aktor dengan fungsi-fungsi utama yang dapat dilakukan pada sistem. Diagram tersebut ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Use Case Diagram

Tabel 4. 4 Deskripsi Use case Diagram

Kode	Kebutuhan Fungsional	Deskripsi
UC01	Membuat Akun	<i>Use case</i> membuat akun menggambarkan proses <i>customer</i> membuat akun baru pada aplikasi.
UC02	Melihat Katalog Buku	<i>Use case</i> melihat katalog buku menggambarkan proses <i>customer</i> melihat, dan melihat informasi lengkap salah satu buku yang tersedia pada aplikasi.

Kode	Kebutuhan Fungsional	Deskripsi
UC03	Mengelola Keranjang	<i>Use case</i> mengelola keranjang menggambarkan proses <i>customer</i> menambahkan, mengubah jumlah, atau menghapus item pada keranjang belanja.
UC04	Mengelola Alamat Pengiriman	<i>Use case</i> mengelola alamat pengiriman menggambarkan proses <i>customer</i> menambah, mengubah, dan menetapkan alamat utama pengguna.
UC05	Melakukan Pemesanan	<i>Use case</i> melakukan pemesanan menggambarkan proses <i>customer</i> membuat pesanan berdasarkan item pada keranjang belanja, mulai dari pemilihan alamat dan layanan pengiriman hingga pengiriman bukti pembayaran atas pesanan yang telah dibuat.
UC06	Melihat Riwayat Pesanan	<i>Use case</i> melihat riwayat dan detail pesanan menggambarkan proses <i>customer</i> melihat daftar pesanan beserta informasi detail dan status pesanan serta menerima notifikasi terkait perubahan status pesanan dan pembayaran.
UC07	Melihat <i>Dashboard</i>	<i>Use case</i> melihat <i>dashboard</i> menggambarkan proses admin melihat ringkasan data penjualan pada sistem.
UC08	Mengelola Pesanan	<i>Use case</i> mengelola pesanan menggambarkan proses admin melihat, memfilter, mengonfirmasi pembayaran, memasukkan nomor resi pesanan, dan mencatat transaksi penjualan tunai yang dilakukan secara langsung, serta menerima notifikasi ketika terdapat pesanan masuk yang memerlukan konfirmasi pembayaran.
UC09	Mengelola Buku	<i>Use case</i> mengelola buku menggambarkan proses admin menambah, mengubah, dan menghapus data buku pada katalog.

Tabel 4. 5 Deskripsi Relasi Use case

No.	Relasi	Jenis Relasi	Deskripsi
1.	Melakukan pemesanan → Mengelola Keranjang	<i>Include</i>	<i>Use case</i> melakukan pemesanan selalu menggunakan data item yang terdapat pada keranjang belanja.
2.	Mengelola Alamat Pengiriman → Melakukan pemesanan	<i>Extend</i>	<i>Use case</i> mengelola alamat pengiriman merupakan perluasan dari <i>use case</i> melakukan pemesanan yang hanya dijalankan apabila <i>customer</i> belum memiliki alamat pengiriman atau perlu menambah maupun mengubah alamat pada saat proses pemesanan berlangsung.

4.2.2 Skenario *Use case*

Skenario *Use case* pada aplikasi toko buku *online* tekMIRA disusun untuk menggambarkan alur interaksi antara aktor dan sistem pada setiap fungsi yang terdapat dalam *Use case diagram*. Penjelasan skenario *Use case* tersebut disajikan pada tabel 4.6 - 4.28.

Tabel 4. 6 Use case Register Akun

Identifikasi	
Kode	UC01
Nama	Membuat Akun

Tujuan	Untuk membuat akun baru agar <i>customer</i> dapat menggunakan fitur aplikasi yang memerlukan autentikasi	
Deskripsi	<i>Use case</i> membuat akun digunakan oleh <i>customer</i> untuk melakukan pendaftaran akun baru pada aplikasi toko buku <i>online</i> tekMIRA	
Aktor	<i>Customer</i>	
Skenario Utama		
Kondisi awal	<i>Customer</i> belum memiliki akun dan berada pada halaman register akun	
Aksi Aktor		Reaksi Sistem
1. Mengisi data registrasi		2. Menerima data registrasi yang dimasukkan oleh <i>customer</i>
3. Menekan tombol daftar		4. Memvalidasi data registrasi
		5. Membuat akun baru apabila data registrasi valid
		6. Mengarahkan <i>customer</i> ke halaman utama aplikasi
Kondisi Akhir	Akun <i>customer</i> berhasil dibuat dan <i>customer</i> dapat menggunakan aplikasi sebagai pengguna terdaftar	
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem
1. Mengisi data registrasi dengan email yang sudah terdaftar atau data yang tidak valid		2. Menampilkan pesan kesalahan

Tabel 4. 7 Use case Melihat Katalog Buku

Identifikasi	
Kode	UC02
Nama	Melihat Katalog Buku
Tujuan	Untuk menampilkan daftar buku yang tersedia pada aplikasi beserta informasi lengkap dari buku yang dipilih pada aplikasi
Deskripsi	<i>Use case</i> melihat katalog buku digunakan oleh <i>customer</i> untuk melihat daftar publikasi teknis yang tersedia, mencari, hingga melihat informasi lengkap buku yang tersedia pada aplikasi.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi awal	<i>Customer</i> membuka aplikasi dan berada pada halaman utama atau halaman katalog buku.
Aksi Aktor	Reaksi Sistem
1. Membuka halaman katalog buku	2. Mengambil data buku yang tersedia
	3. Menampilkan daftar buku pada halaman katalog
4. Memasukkan kata kunci pencarian	5. Menampilkan daftar buku sesuai kata kunci yang dimasukkan
6. Memilih salah satu buku dari daftar	7. Mengambil data detail buku yang dipilih
	8. Menampilkan halaman detail buku yang dipilih
Kondisi Akhir	Daftar buku yang tersedia berhasil ditampilkan kepada <i>Customer</i> .

Tabel 4. 8 Use case Mengelola Keranjang

Identifikasi

Kode	UC03
Nama	Mengelola Keranjang
Tujuan	Untuk mengelola item buku yang akan dibeli oleh <i>Customer</i> .
Deskripsi	<i>Use case</i> mengelola keranjang digunakan oleh <i>customer</i> untuk menambahkan buku, mengubah jumlah item, atau menghapus item pada keranjang belanja.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi awal	<i>Customer</i> telah <i>login</i> dan berada pada halaman buku atau halaman keranjang.
Aksi Aktor	
Reaksi Sistem	
1. Menambahkan buku ke keranjang	2. Menyimpan item buku ke dalam keranjang belanja
3. Membuka halaman keranjang	4. Menampilkan daftar item pada keranjang belanja
5. Mengubah jumlah atau menghapus item keranjang	6. Memperbarui data keranjang belanja
	7. Menghitung ulang total harga belanja
Kondisi Akhir	Data keranjang belanja <i>Customer</i> berhasil diperbarui.

Tabel 4. 9 Use case Mengelola Alamat Pengiriman

Identifikasi	
Kode	UC04
Nama	Mengelola Alamat Pengiriman
Tujuan	Untuk mengelola data alamat pengiriman yang digunakan dalam proses pembelian.
Deskripsi	Use case mengelola alamat pengiriman digunakan oleh <i>customer</i> untuk menambah, mengubah, menghapus, dan menetapkan alamat utama.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi awal	<i>Customer</i> telah <i>login</i> dan berada pada halaman alamat pengiriman.
Aksi Aktor	Reaksi Sistem
1. Menambah, menghapus, atau menetapkan alamat utama	2. Memvalidasi data alamat yang diproses
	3. Menyimpan perubahan data alamat pengiriman
	4. Menampilkan daftar alamat yang telah diperbarui
Kondisi Akhir	Data alamat pengiriman <i>customer</i> berhasil diperbarui.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
83. Mengisi data alamat secara tidak lengkap	84. Menampilkan pesan kesalahan dan data alamat tidak tersimpan

Tabel 4. 10 Use case Melakukan Pesanan

Identifikasi	
Kode	UC05
Nama	Melakukan Pesanan

Tujuan	Untuk membuat pesanan berdasarkan item buku yang terdapat pada keranjang belanja hingga mengirimkan bukti pembayaran atas pesanan tersebut.		
Deskripsi	Use case melakukan pesanan digunakan oleh customer untuk membuat pesanan dengan memilih item keranjang, alamat pengiriman, dan kurir yang tersedia, kemudian mengunggah bukti pembayaran atas pesanan yang telah dibuat.		
Aktor	Customer		
Skenario Utama			
Kondisi awal	Customer telah login, berada di halaman keranjang dan memiliki item pada keranjang belanja.		
Aksi Aktor		Reaksi Sistem	
1. Memilih item yang akan dibeli dan menekan tombol checkout		2. Menerima data item keranjang yang di-checkout	
		3. Menampilkan halaman checkout pesanan	
4. Memilih alamat pengiriman		5. Menghitung ongkos kirim berdasarkan alamat dan berat pesanan	
6. Memilih layanan kurir		7. Menampilkan ringkasan pesanan dan total pembayaran	
8. Menekan tombol buat pesanan		9. Membuat data pesanan baru dengan status pending_payment	
		10. Mengarahkan customer ke halaman pembayaran	
11. Memilih file bukti transfer dan menekan tombol unggah		12. Menunggah bukti pembayaran ke server	
		13. Memperbarui status pesanan menjadi waiting_confirmation	
		14. Mengirimkan notifikasi konfirmasi pembayaran ke admin	
Kondisi Akhir	Pesanan berhasil dibuat dan bukti pembayaran berhasil diunggah, sehingga pesanan berstatus waiting_confirmation dan menunggu konfirmasi admin.		
Skenario Alternatif			
Aksi Aktor		Reaksi Sistem	
1. Melakukan proses checkout tanpa memiliki alamat pengiriman tersimpan		2. Meminta menambahkan alamat pengiriman terlebih dahulu sebelum melakukan pemesanan	
3. Menunda pengunggahan bukti pembayaran setelah pesanan berhasil dibuat		4. Menyimpan pesanan dengan status pending_payment	
5. Menunggah bukti pembayaran yang gagal diproses		6. Menampilkan pesan kesalahan dan status pesanan tidak diperbarui	

Tabel 4. 11 Use case Melihat Riwayat Pesanan

Identifikasi	
Kode	UC06
Nama	Melihat Riwayat Pesanan
Tujuan	Untuk menampilkan riwayat pesanan dan informasi detail status pesanan <i>Customer</i> .
Deskripsi	<i>Use case</i> melihat riwayat dan detail pesanan digunakan oleh <i>Customer</i> untuk melihat daftar pesanan, detail pesanan, status pembayaran, status pengiriman, dan nomor resi apabila tersedia.
Aktor	<i>Customer</i>
Skenario Utama	
Kondisi awal	<i>Customer</i> telah <i>login</i> dan berada pada halaman pesanan.

Aksi Aktor		Reaksi Sistem
1. Memilih salah satu pesanan		2. Menampilkan detail pesanan yang dipilih
		3. Menampilkan status pesanan dan informasi pengiriman apabila tersedia
4. Menekan tombol konfirmasi apabila pesanan berstatus SHIPPED		5. Memperbarui status pesanan menjadi COMPLETED
Kondisi Akhir	Riwayat dan detail pesanan berhasil ditampilkan kepada <i>Customer</i> .	

Tabel 4. 12 Use case Melihat Dashboard

Identifikasi	
Kode	UC07
Nama	Melihat <i>Dashboard</i>
Tujuan	Untuk menampilkan ringkasan data penjualan kepada Admin.
Deskripsi	<i>Use case</i> melihat <i>dashboard</i> digunakan oleh Admin untuk melihat statistik penjualan, grafik penjualan bulanan, dan daftar buku terlaris.
Aktor	Admin
Skenario Utama	
Kondisi awal	Admin telah <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Membuka halaman <i>dashboard</i>	2. Mengambil data ringkasan penjualan
	3. Menampilkan statistik penjualan, grafik penjualan bulanan dan daftar buku terlaris
4. Menekan penjualan bulan tertentu	5. Mengambil data ringkasan penjualan pada bulan yang dipilih
	6. Menampilkan detail penjualan bulan yang dipilih
Kondisi Akhir	Ringkasan data penjualan berhasil ditampilkan kepada Admin.

Tabel 4. 13 Use case Mengelola Pesanan

Identifikasi	
Kode	UC08
Nama	Mengelola Pesanan
Tujuan	Untuk mengelola status pesanan dan proses pengiriman pesanan <i>Customer</i> , serta mencatat transaksi penjualan tunai yang dilakukan secara langsung
Deskripsi	<i>Use case</i> mengelola pesanan digunakan oleh Admin untuk melihat daftar pesanan, memfilter pesanan, mengonfirmasi pembayaran, memasukkan nomor resi pengiriman, dan mencatat transaksi penjualan tunai yang terjadi di luar aplikasi.
Aktor	Admin
Skenario Utama	
Kondisi awal	Admin telah <i>login</i> dan berada pada halaman pengelolaan pesanan.
Aksi Aktor	Reaksi Sistem
1. Memfilter pesanan berdasarkan status	2. Menampilkan daftar pesanan sesuai status yang dipilih
3. Membuka detail pesanan	4. Menampilkan informasi detail pesanan

5. Mengonfirmasi pembayaran atau memasukkan nomor resi pengiriman	6. Memperbarui status pesanan dan menyimpan nomor resi apabila tersedia
	7. Mengirimkan notifikasi terkait pembaruan status kepada <i>customer</i>
1. Menekan tombol jual langsung untuk mencatat transaksi tunai	2. Menampilkan formulir transaksi penjualan tunai
3. Mengisi data pembeli dan memilih buku yang dibeli	4. Menghitung total harga transaksi
5. Memproses transaksi penjualan tunai	6. Menyimpan data transaksi sebagai pesanan baru
	7. Memperbarui stok buku yang terjual
Kondisi Akhir	Status pesanan berhasil diperbarui oleh Admin, atau transaksi penjualan tunai berhasil dicatat dan stok buku diperbarui.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Memasukkan data nomor resi yang tidak valid atau kosong pada saat diperlukan	2. Menampilkan pesan kesalahan dan perubahan status tidak disimpan
3. Memasukkan jumlah pembelian melebihi stok yang tersedia pada transaksi tunai	4. Menampilkan pesan bahwa stok buku tidak mencukupi dan transaksi tidak diproses

Tabel 4. 14 Use case Mengelola Data Buku

Identifikasi	
Kode	UC09
Nama	Mengelola Data Buku
Tujuan	Untuk mengelola data buku yang ditampilkan pada katalog aplikasi.
Deskripsi	Use case mengelola data buku digunakan oleh Admin untuk menambah, mengubah, dan menghapus data buku beserta gambar buku.
Aktor	Admin
Skenario Utama	
Kondisi awal	Admin telah <i>login</i> dan berada pada halaman pengelolaan data buku.
Aksi Aktor	Reaksi Sistem
1. Menambah, mengubah atau menghapus data buku	2. Menampilkan formulir atau konfirmasi sesuai aksi yang dipilih
1. Menyimpan perubahan data buku	2. Memvalidasi data buku
	3. Menyimpan perubahan data buku pada server
	4. Menampilkan daftar buku yang telah diperbarui
Kondisi Akhir	Data buku pada katalog berhasil diperbarui.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Mengisi data buku secara tidak lengkap	2. Menampilkan pesan kesalahan dan data buku tidak disimpan

4.3 Perancangan Kelas

4.3.1 Mengidentifikasi Kelas-Kelas Analisis

Identifikasi kelas analisis dilakukan untuk menentukan kelas yang berperan dalam proses kerja aplikasi toko buku online tekMIRA. Kelas analisis disusun berdasarkan pendekatan Boundary, Control, dan Entity.

Tabel 4. 15 Kelas - Kelas Analisis

No.	Nama Kelas	Jenis	Deskripsi
1.	User	Entity	Merepresentasikan data pengguna aplikasi.
2.	Book	Entity	Merepresentasikan data utama buku yang dijual pada aplikasi.
3.	BookImage	Entity	Merepresentasikan data gambar tambahan yang dimiliki suatu buku.
4.	CartItem	Entity	Merepresentasikan item buku yang terdapat dalam keranjang.
5.	Address	Entity	Merepresentasikan data alamat pengguna untuk keperluan pengiriman.
6.	Order	Entity	Merepresentasikan data transaksi pesanan pengguna.
7.	OrderItem	Entity	Merepresentasikan detail buku yang terdapat dalam suatu pesanan.
8.	ShippingOption	Entity	Merepresentasikan pilihan layanan pengiriman yang tersedia.
9.	AppNotification	Entity	Merepresentasikan data notifikasi aplikasi.
10.	RegisterPage	Boundary	Menampilkan formulir pembuatan akun pengguna.
11.	BookListPage	Boundary	Menampilkan katalog buku yang tersedia.
12.	BookDetailPage	Boundary	Menampilkan informasi lengkap buku.
13.	CartPage	Boundary	Menampilkan dan menyediakan pengelolaan isi keranjang belanja.
14.	AdressListPage	Boundary	Menampilkan daftar alamat dan pengelolaan alamat pengguna.
15.	AddressFormSheet	Boundary	Menampilkan formulir untuk menambah dan mengubah alamat pengiriman.
16.	CheckoutPage	Boundary	Menampilkan proses pemesanan, pemilihan alamat, dan layanan pengiriman.
17.	PaymentPage	Boundary	Menampilkan informasi pembayaran dan fasilitas unggah bukti pembayaran.
18.	OrderPage	Boundary	Menampilkan daftar riwayat pesanan pengguna.
19.	OrderDetailPage	Boundary	Menampilkan detail dan status pesanan.
20.	NotificationPage	Boundary	Menampilkan daftar notifikasi aplikasi kepada pengguna
21.	AdminHomePage	Boundary	Menampilkan ringkasan data penjualan pada dashboard admin.
22.	AdminOrdersPage	Boundary	Menampilkan daftar pesanan yang dikelola oleh admin
23.	AdminOrderDetailPage	Boundary	Menampilkan detail pesanan serta proses pengelolaan pesanan oleh admin
24.	AdminCashSalePage	Boundary	Menampilkan proses pencatatan transaksi penjualan tunai oleh admin
25.	AdminBooksPage	Boundary	Menampilkan daftar dan pengelolaan data buku oleh admin
26.	AdminBookFormPage	Boundary	Menampilkan formulir penambahan dan perubahan data buku

No.	Nama Kelas	Jenis	Deskripsi
27.	RegisterController	Control	Mengelola proses pembuatan akun pengguna.
28.	BooksCatalogController	Control	Mengelola pemuatan dan pengolahan data katalog buku.
29.	CartController	Control	Mengelola penambahan, perubahan, jumlah, dan penghapusan item keranjang.
30.	AddressController	Control	Mengelola data alamat pengiriman pengguna.
31.	CheckoutController	Control	Mengelola proses checkout dan pembuatan pesanan.
32.	OrdersController	Control	Mengelola pemuatan daftar dan detail pesanan pengguna.
33.	AdminDashboardController	Control	Mengelola data ringkasan penjualan untuk dashboard admin.
34.	AdminOrdersController	Control	Mengelola pesanan pada sisi admin, termasuk konfirmasi pembayaran dan pengisian nomor resi
35.	AdminBooksController	Control	Mengelola data buku pada sisi admin

4.3.2 Menentukan Atribut-Atribut

Atribut kelas ditentukan untuk menggambarkan data yang dimiliki oleh setiap kelas pada aplikasi toko buku *online* tekMIRA. Penentuan atribut dilakukan berdasarkan kebutuhan fungsional sistem yang telah dianalisis, sehingga atribut yang dicantumkan hanya atribut yang berperan dalam proses bisnis utama.

Tabel 4. 16 Atribut - Atribut Kelas

No	Nama Kelas	Jenis	Atribut
1.	User	Entity	id, email, role, username, name, phoneNumber, profilePicture, studentStatus, institution, major, studentIdentityNumber, studentIdCardPath, dateOfBirth, createdAt, updatedAt
2.	Book	Entity	id, title, imageUrl, price, stock, soldCount, averageRating, reviewCount, description
3.	BookImage	Entity	id, bookId, imagePath, createdAt, updatedAt
4.	CartItem	Entity	id, bookId, title, price, quantity, stock, weight, imageUrl
5.	Order	Entity	id, status, shippingAddress, shippingCost, subtotal, totalPrice, discountAmount, discountName, discountPercentage, courierService, courierServiceType, trackingNumber, paymentProof, paymentDeadline, createdAt, items
6.	OrderItem	Entity	id, bookId, title, imageUrl, qty, priceAtPurchase, weight
7.	Address	Entity	id, label, recipientName, phoneNumber, streetAddress, city, province, postalCode, cityId, isPrimary, latitude, longitude
8.	AppNotification	Entity	id, type, title, message, orderId, isRead, createdAt
9.	ShippingOption	Entity	id, name, cost, eta
10.	RegisterPage	Boundary	username, email, password, confirmPassword, fullName
11.	BooksListPage	Boundary	searchQuery, selectedSort
12.	BookDetailPage	Boundary	bookId
13.	CartPage	Boundary	selectedIds
14.	CheckoutPage	Boundary	selectedCartItemIds, selectedAddress, selectedShipping, discountCode

No	Nama Kelas	Jenis	Atribut
15.	PaymentPage	Boundary	orderId
16.	OrdersPage	Boundary	selectedStatus
17.	OrderDetailPage	Boundary	orderId
18.	AddressListPage	Boundary	selectedAddress
19.	AddressFormSheet	Boundary	label, recipientName, phoneNumber, streetAddress, selectedCity, selectedProvince, postalCode
20.	NotificationPage	Boundary	selectedNotification
21.	AdminHomePage	Boundary	selectedYear
22.	AdminOrdersPage	Boundary	selectedStatus, selectedOrderType
23.	AdminOrderDetailPage	Boundary	orderId, trackingNumber
24.	AdminBooksPage	Boundary	searchKeyword
25.	AdminBookFormPage	Boundary	bookId, title, description, price, weight, stock, coverImage, additionalImages
26.	AdminCashSalePage	Boundary	customerName, notes, date, paymentMethod, paymentProof, cashSaleItems
27.	RegisterController	Control	authRepository, secureStorage, currentUser
28.	BooksCatalogController	Control	booksRepository, reviewRepository, bookId, bookImages, searchQuery, selectedSort, page, hasMore
29.	CartController	Control	cartRepository, cartItems
30.	AddressController	Control	Dio, addresses, destinations
31.	CheckoutController	Control	orderRepository, cartRepository, selectedAddress, selectedShipping, shippingOptions, orderData, orderId, paymentProof
32.	OrdersController	Control	orderRepository, orderId, currentStatus, page, hasMore, orderDetail
33.	NotificationController	Control	notificationRepository, notifications, firebaseMessaging, fcmToken
34.	AdminDashboardController	Control	reportsRepository, selectedYear, salesSummary, monthlySummary, bestSellers
35.	AdminOrdersController	Control	adminOrderRepository, orderId, selectedStatus, selectedOrderType, cashSaleItems
36.	AdminBooksController	Control	adminBooksRepository, bookId, searchQuery, page, hasMore, formState

4.3.3 Mendefinisikan Operasi-Operasi

Setiap objek memiliki tanggung jawab atau perilaku tertentu yang dinyatakan melalui operasinya. Operasi-operasi ini dapat dilihat pada tabel 4.17

Tabel 4. 17 Operasi - Operasi Kelas

No	Nama Kelas	Kandidat Perilaku	Nama Perilaku
1.	User	Membentuk objek pengguna dari data JSON.	fromJson(json)
		Menampilkan tanggal lahir pengguna dalam format tampilan.	dateOfBirthDisplay()
2.	Book	Membentuk objek buku dari data JSON.	fromJson(json)
3.	BookImage	Membentuk objek gambar buku dari data JSON.	fromJson(json)
		Menghasilkan URL gambar buku.	url()
4.	CartItem	Membentuk item keranjang dari respons API.	fromApi(json)

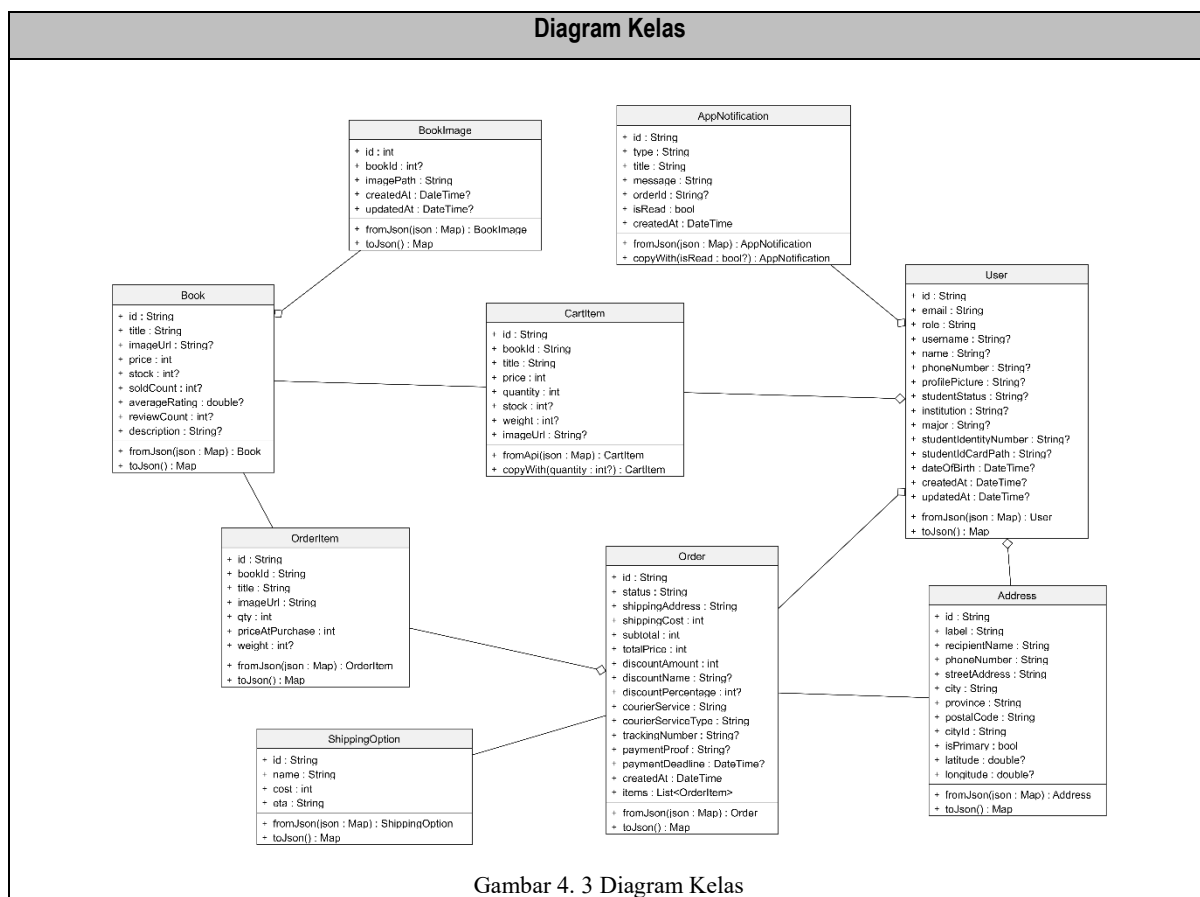
No	Nama Kelas	Kandidat Perilaku	Nama Perilaku
		Membuat salinan item keranjang dengan perubahan tertentu.	copyWith()
5.	Order	Membentuk objek pesanan dari data JSON.	fromJson(json)
6.	OrderItem	Membentuk item pesanan dari data JSON.	fromJson(json)
7.	Address	Membentuk objek alamat dari data JSON.	fromJson(json)
8.	AppNotification	Membentuk objek notifikasi dari data JSON.	fromJson(json)
		Membuat salinan notifikasi dengan perubahan status baca.	copyWith()
9.	ShippingOption	Membentuk objek opsi pengiriman dari data JSON.	fromJson(json)
10.	RegisterPage	Mengirim data registrasi akun baru ke sistem.	submitRegister()
11.	BooksListPage	Menampilkan daftar buku pada halaman katalog utama.	showBookCatalog()
12.	BookDetailPage	Menampilkan detail informasi buku berdasarkan buku yang dipilih.	showBookDetail()
		Menambahkan buku ke keranjang berdasarkan jumlah yang dipilih.	addToCart(quantity)
		Membuat proses pembelian langsung dari halaman detail buku.	buyNow(quantity)
		Menambahkan atau menghapus buku dari wishlist.	toggleWishlist()
13.	CartPage	Menghapus item yang dipilih dari keranjang.	removeSelected()
		Melanjutkan item keranjang yang dipilih ke proses checkout.	checkoutSelectedItems()
14.	CheckoutPage	Membuat pesanan berdasarkan item, alamat, pengiriman, dan diskon yang dipilih.	createOrder()
15.	PaymentPage	Menampilkan informasi pembayaran berdasarkan pesanan.	showPaymentDetail(), uploadPaymentProof()
16.	OrdersPage	Menampilkan daftar pesanan berdasarkan status yang dipilih.	showOrdersByStatus()
17.	OrderDetailPage	Membatalkan pesanan pengguna.	cancelOrder()
		Menandai pesanan sebagai selesai.	markOrderCompleted()
18.	AddressListPage	Menghapus alamat pengiriman pengguna.	deleteAddress()
		Menentukan alamat sebagai alamat utama.	setPrimaryAddress()
19.	AddressFormSheet	Menyimpan data alamat baru atau perubahan alamat.	submitAddress()
20.	NotificationPage	Menandai satu notifikasi sebagai sudah dibaca.	markAsRead()
		Menandai seluruh notifikasi sebagai sudah dibaca.	markAllAsRead()
21.	AdminHomePage	Memilih tahun laporan pada dashboard admin.	selectYear()
22.	AdminOrdersPage	Menampilkan daftar pesanan admin berdasarkan status dan tipe pesanan.	showAdminOrders()
23.	AdminOrderDetailPage	Mengonfirmasi pembayaran pesanan.	confirmPayment()
		Menyimpan nomor resi pengiriman.	saveShipment()
24.	AdminBooksPage	Menghapus data buku dari daftar buku admin.	deleteBook()
25.	AdminBookFormPage	Menyimpan data buku baru atau perubahan data buku.	submitBookForm()

No	Nama Kelas	Kandidat Perilaku	Nama Perilaku
26.	AdminCashSalePage	Memproses transaksi penjualan tunai oleh admin.	processCashSale()
27.	RegisterController	Membuat akun pengguna baru.	register(username, email, password, fullName)
28.	BookCatalogController	Memuat ulang data buku dari awal	Refresh()
		Memuat halaman berikutnya pada katalog buku	fetchNext()
29.	BookDetailController	Mengambil detail data buku	getBookDetail(bookId)
		Mengambil gambar tambahan buku	getBookImages(bookId)
		Mengambil rekomendasi buku	getBookRecommendations(bookId)
		Memuat ulang daftar ulasan buku	refresh(force)
		Memuat halaman berikutnya pada daftar ulasan	fetchNext()
		Mengatur filter ulasan berdasarkan rating	setStarFilter(star)
		Mengatur urutan tampilan ulasan	setSort(sort)
30.	CartController	Menambahkan buku ke keranjang	add(bookId, quantity)
		Mengubah jumlah item pada keranjang	changeQty(cartItemId, quantity)
		Menghapus item dari keranjang	Remove(cartItemId)
		Menyiapkan keranjang untuk pembelian langsung	buyNowEnsureSingle(bookId, quantity)
		Mengosongkan data keranjang	clear()
31.	AddressController	Menambahkan alamat pengiriman baru	create(payload)
		Memperbarui data alamat pengiriman	update(id, payload)
		Menghapus alamat pengiriman	delete(id)
		Menjadikan alamat sebagai alamat utama	setPrimary(id)
		Mencari kota atau destinasi pengiriman	search(keyword)
32.	CheckoutController	Memilih alamat pengiriman.	pickAddress(id)
		Memilih layanan pengiriman beserta biaya.	pickShippingWithCost(id, cost)
		Menghapus pilihan pengiriman.	clearShipping()
		Mengatur kode diskon.	setDiscount(code)
		Mengambil pilihan layanan pengiriman berdasarkan alamat dan berat pesanan.	getShippingOptions(query)
		Membuat pesanan baru.	createOrder(payload, cartItemIdsForCleanup)
		Mengunggah bukti pembayaran.	upload(orderId, filePath)
33.	OrdersController	Memuat daftar pesanan berdasarkan status.	load(refresh, status)
		Mengambil detail pesanan berdasarkan id pesanan	getOrderDetail(orderId)
34.	NotificationController	Memuat ulang daftar notifikasi.	refresh()
		Menambahkan notifikasi yang diterima dari layanan pesan.	addFromSocket(notification)
		Menandai satu notifikasi sebagai dibaca.	markAsRead(id)
		Menandai seluruh notifikasi sebagai dibaca.	markAllAsRead()
		Mengambil token Firebase Cloud Messaging perangkat	getFCMToken()
		Menghubungkan layanan notifikasi dengan sesi pengguna	connect(token)
35.	AdminDashboardController	Memuat ringkasan penjualan.	getSalesSummary(year)

No	Nama Kelas	Kandidat Perilaku	Nama Perilaku
		Memuat data penjualan bulanan.	getMonthlySummary(year)
		Memuat daftar buku terlaris.	getBestsellers(year)
36.	AdminOrdersController	Memuat daftar pesanan admin berdasarkan filter.	load(status, orderType)
		Mengkonfirmasi penerimaan pembayaran dari pengguna	confirmPayment(orderId)
		Menambahkan resi pengiriman	shipOrder(orderId, tracking)
		Menambahkan item buku ke transaksi tunai.	addItem(book)
		Mengubah jumlah item transaksi tunai.	updateQuantity(bookId, quantity)
		Mengubah harga jual khusus item transaksi tunai.	updateCustomPrice(bookId, price)
		Menghapus item dari transaksi tunai.	removeItem(bookId)
		Mengosongkan daftar item transaksi tunai.	clear()
37.	AdminBooksController	Mengatur kata kunci pencarian buku.	setSearch(query)
		Memuat daftar buku pada sisi admin.	load(refresh, query)
		Memuat halaman berikutnya pada daftar buku admin.	loadMore()
		Memuat data buku untuk proses ubah data.	loadForEdit(id)
		Mengatur nilai atribut pada formulir buku.	setField()
		Memilih gambar sampul buku.	pickCover()
		Memilih gambar tambahan buku.	pickAdditionalImages()
		Menghapus gambar buku yang sudah tersimpan.	removeExistingImage(imageId)
		Menyimpan data buku baru.	submitCreate()
		Memperbarui data buku.	submitUpdate(id)

4.3.4 Perancangan Kelas

Perancangan kelas dilakukan dengan menyusun diagram kelas berdasarkan kelas-kelas analisis yang telah diidentifikasi, dengan fokus pada struktur data domain aplikasi. Diagram kelas digunakan untuk menggambarkan struktur statis model data pada aplikasi toko buku *online* tekMira, mencakup atribut, operasi, dan relasi antar kelas entity, sekaligus berfungsi sebagai pemodelan data aplikasi.



Deskripsi

Gambar 4.3 menggambarkan struktur sembilan kelas *entity* pada Aplikasi Toko Buku Online tekMira beserta relasi antar kelasnya. Kelas *User* menjadi pusat relasi dengan hubungan agregasi *one to many* terhadap *AppNotification*, *Address*, *CartItem*, dan *Order*, yang berarti satu pengguna dapat menerima banyak notifikasi, menyimpan banyak alamat pengiriman, memiliki banyak item keranjang, serta melakukan banyak pesanan.

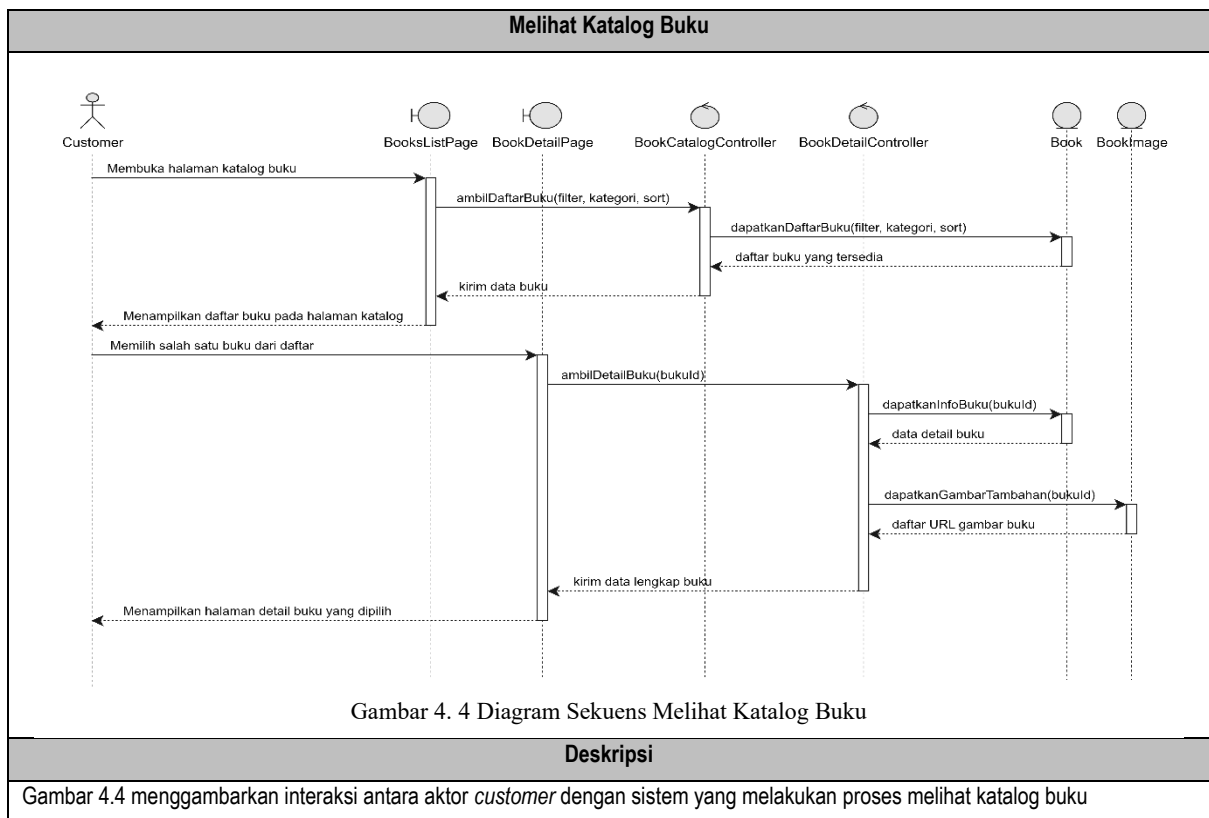
Kelas *Book* memiliki hubungan agregasi *one to many* terhadap *BookImage*, sehingga satu buku dapat memiliki banyak gambar tambahan. Kelas *CartItem* dan *OrderItem* masing-masing merujuk tepat satu *Book*, sedangkan satu *Book* dapat muncul pada banyak item keranjang dan item pesanan.

Kelas *Order* memiliki hubungan komposisi dengan *OrderItem* dengan kardinalitas satu ke satu atau lebih, sehingga setiap pesanan minimal memuat satu item dan item pesanan tidak dapat berdiri tanpa pesanan induknya. Setiap *Order* dikirim ke satu *Address* dan memilih satu *ShippingOption* yang menentukan biaya serta estimasi pengiriman.

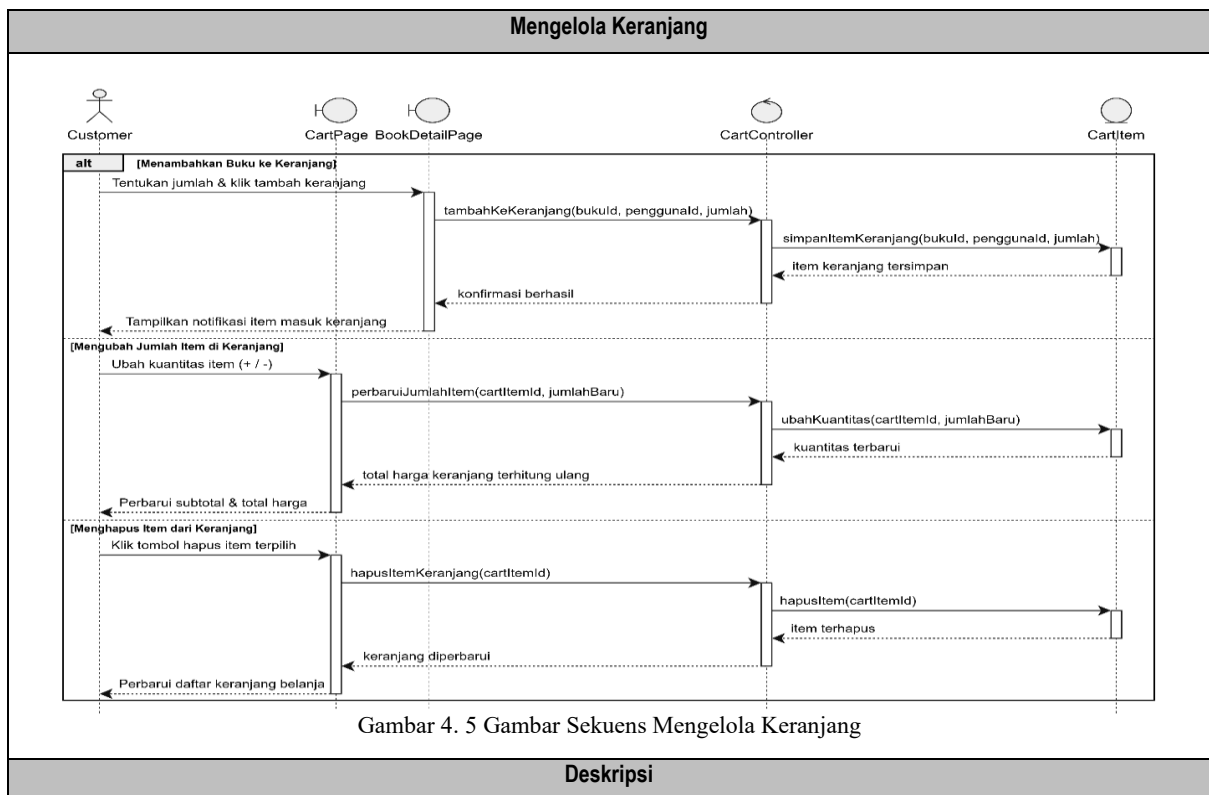
4.3.5 Diagram Sekuens

Pemodelan berbasis perilaku menunjukkan bagaimana sistem atau perangkat lunak akan merespons ketika menerima berbagai kejadian dari luar. Perilaku tersebut dapat diilustrasikan melalui diagram sekuens, seperti yang ditampilkan pada Gambar 4.4 - 4.10

1. Diagram sekuens melihat katalog buku

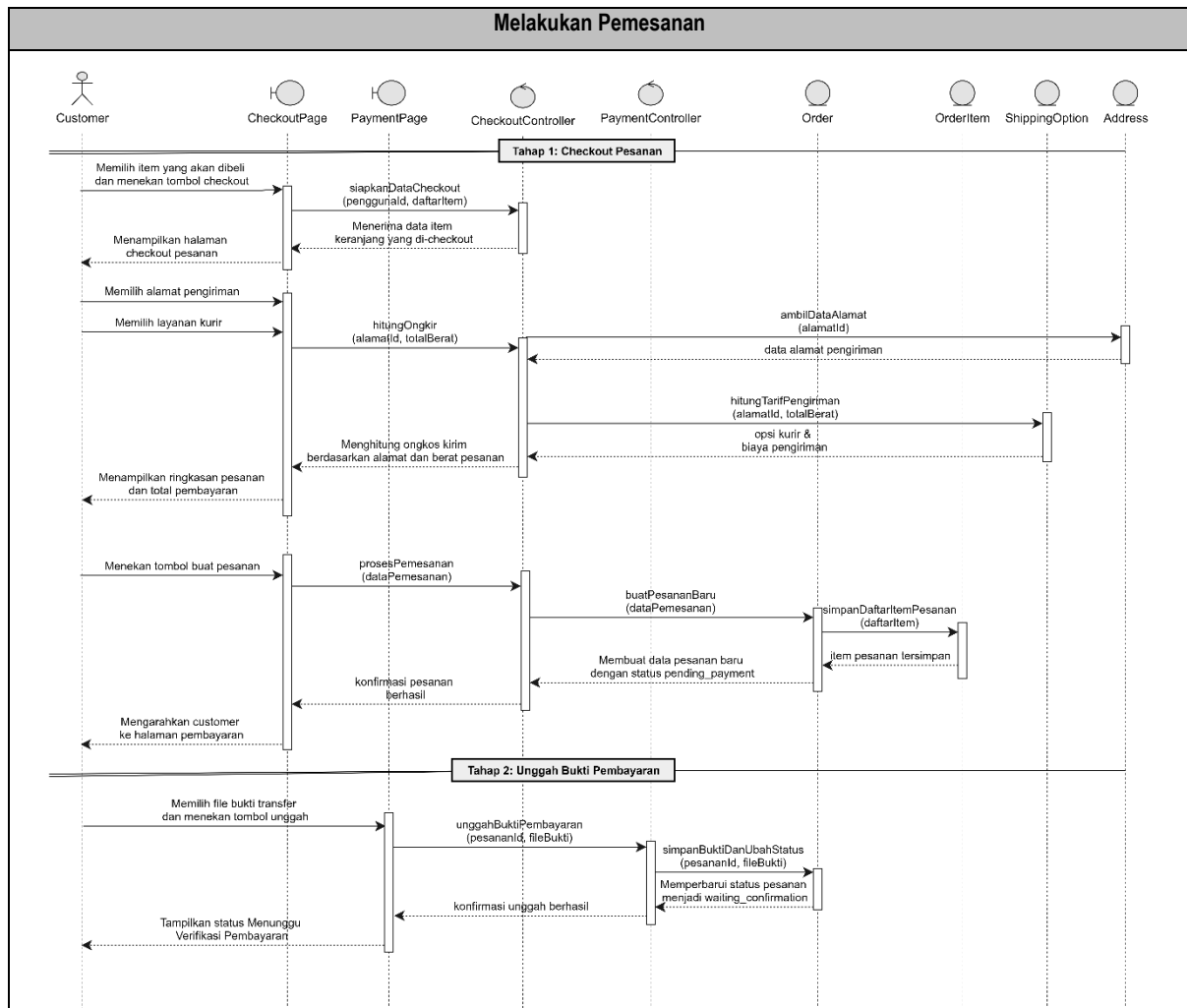


2. Diagram sekuens mengelola keranjang



Gambar 4.5 menggambarkan interaksi antara aktor *customer* dengan sistem yang melakukan proses mengelola keranjang mulai dari menambahkan buku ke keranjang, mengubah jumlah item di keranjang, dan menghapus item dari keranjang.

3. Diagram sekuens melakukan pemesanan

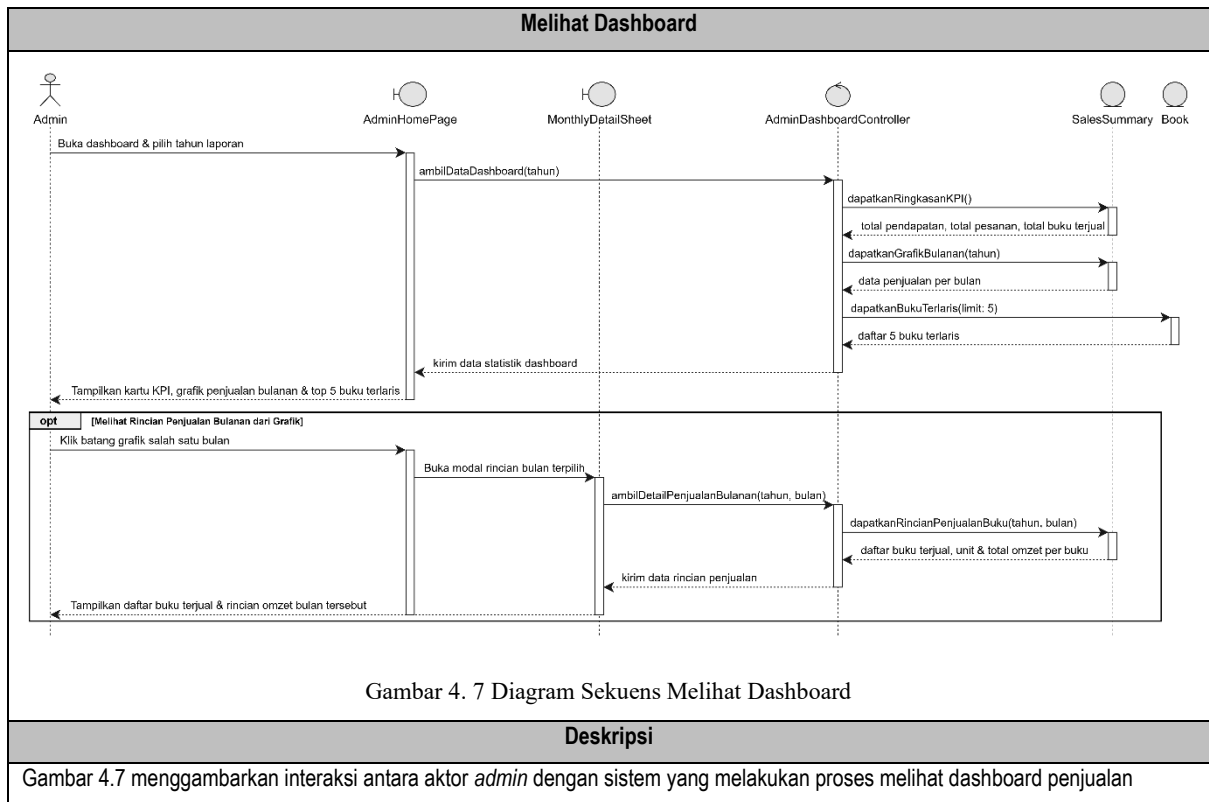


Gambar 4. 6 Diagram Sekuens Melakukan Pemesanan

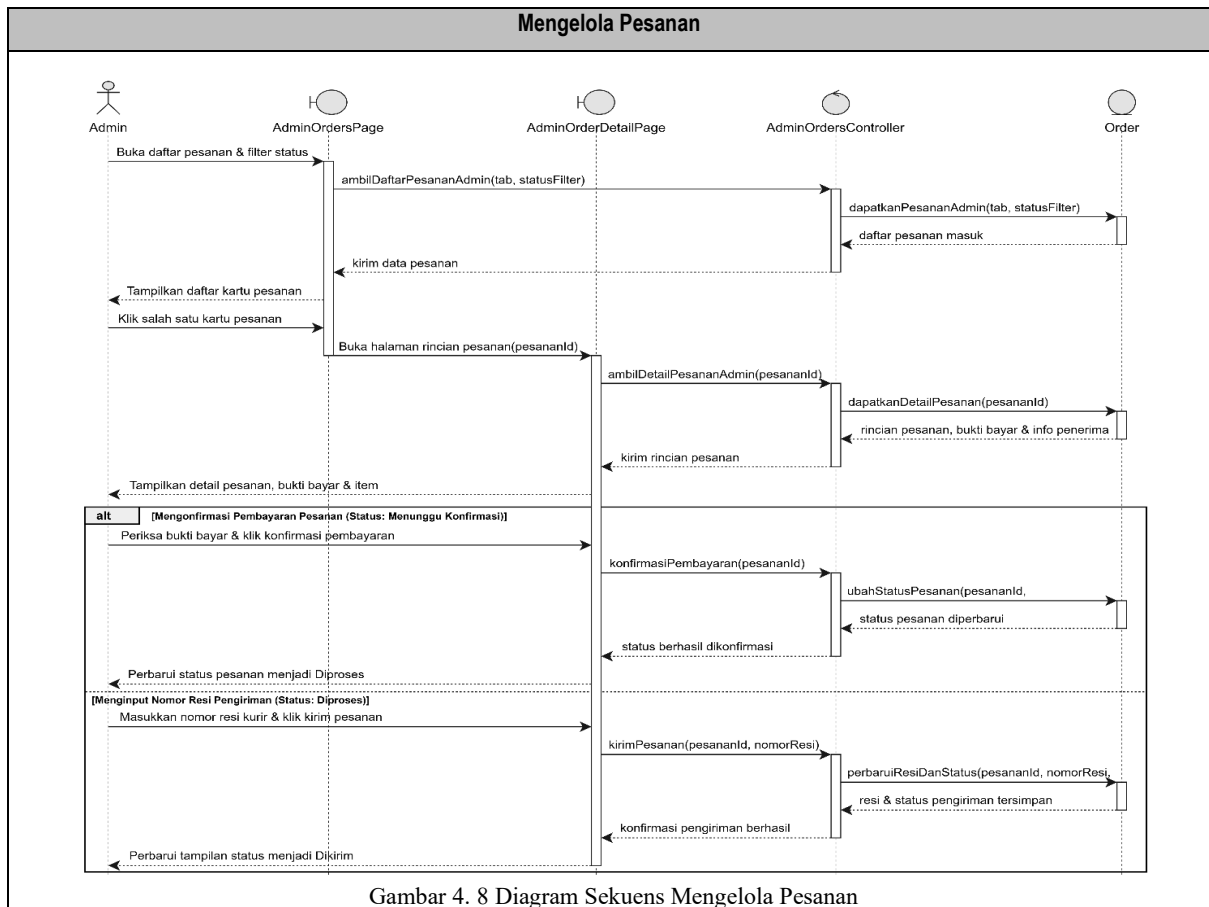
Deskripsi

Gambar 4.6 menggambarkan interaksi antara aktor *customer* dengan sistem yang melakukan proses melakukan pemesanan yang dimulai dari tahap checkout pesanan hingga unggah bukti pembayaran

4. Diagram sekuens melihat dashboard

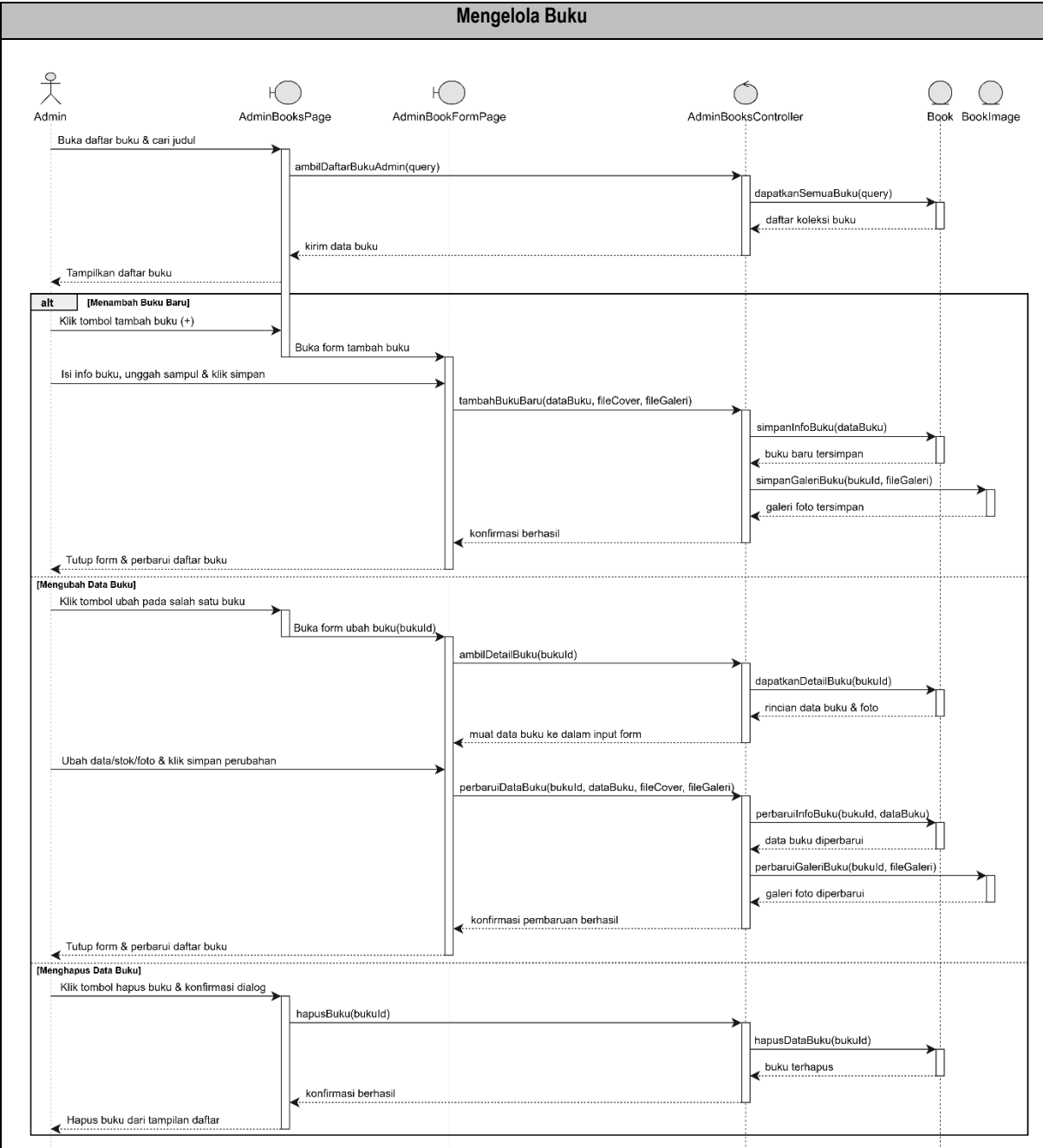


5. Diagram sekuens mengelola pesanan



Deskripsi
Gambar 4.8 menggambarkan interaksi antara aktor <i>admin</i> dengan sistem yang melakukan proses pengelolaan pesanan, yaitu konfirmasi pembayaran dan penginputan nomor resi pengiriman, yang masing-masing memperbarui status pesanan pada Entity Order.

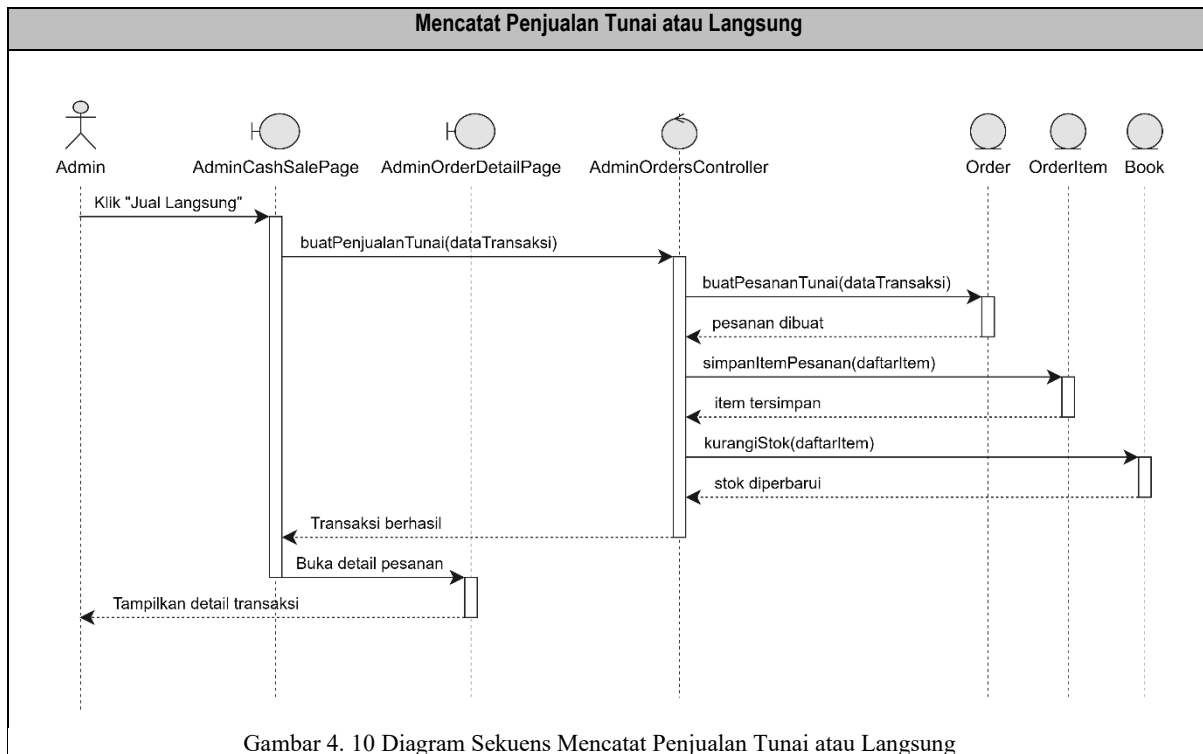
6. Diagram sekuens mengelola buku



Gambar 4. 9 Diagram Sekuens Mengelola Buku

Deskripsi
Gambar 4.9 menampilkan diagram sekuens use case Mengelola Buku yang melibatkan AdminBooksPage, AdminBookFormPage, AdminBooksController, Book, dan BookImage. Diagram menunjukkan alur penambahan, perubahan, dan penghapusan data buku beserta gambar, kemudian memperbarui daftar buku pada halaman Admin.

7. Diagram sekuens mencatat penjualan tunai atau langsung



Deskripsi

Gambar 4. 10 menampilkan diagram sekuens *use case* Mencatat Penjualan Tunai yang melibatkan AdminCashSalePage, BookSearchSheet, AdminOrderDetailPage, AdminCashSaleController, Book, CashSaleItem, dan 'Order'. Diagram menunjukkan alur pencatatan transaksi tunai mulai dari pemilihan buku, pengisian data transaksi, hingga pembuatan pesanan dan pengurangan stok buku.

4.4 Perancangan Perangkat Lunak

4.4.1 Perancangan Data

Tahap pembangunan data bertujuan untuk merinci objek data dan atribut data yang telah dirumuskan pada tahap pemodelan data sebelumnya pada sub-bab 4.3, termasuk hubungan antar kelas *entity* yang digambarkan pada Gambar 4.3. Perincian tersebut mencakup penentuan tipe data dan panjang (*length*) untuk setiap atribut, sehingga struktur data yang dirancang dapat diimplementasikan secara konsisten pada sisi aplikasi. Hasil perancangan tersebut disajikan pada Tabel 4.18.

Tabel 4. 18 Perancangan Data

No.	Objek Data	Atribut Data	Tipe Data	Length	Keterangan
1.	User	Id	String	36	Atribut yang digunakan sebagai ID unik pengguna.
		email	String	50	Atribut yang digunakan untuk menyimpan alamat email pengguna.
		role	String	-	Atribut yang digunakan untuk menyimpan peran pengguna, yaitu user atau admin.
		username	String	30	Atribut yang digunakan untuk menyimpan nama pengguna.
		name	String	100	Atribut yang digunakan untuk menyimpan nama lengkap pengguna.
		phoneNumber	String	20	Atribut yang digunakan untuk menyimpan nomor telepon pengguna.

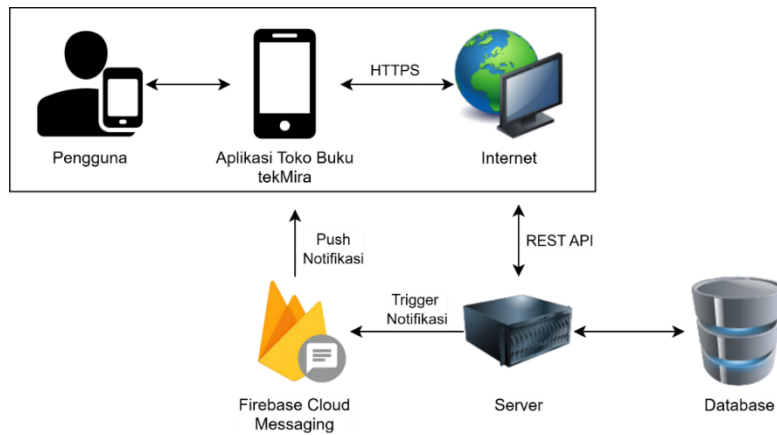
No.	Objek Data	Atribut Data	Tipe Data	Length	Keterangan
		profilePicture	String	255	Atribut yang digunakan untuk menyimpan URL foto profil pengguna.
		studentStatus	String	20	Atribut yang digunakan untuk menyimpan status verifikasi mahasiswa pengguna.
		instution	String	100	Atribut yang digunakan untuk menyimpan nama institusi pendidikan pengguna.
		major	String	100	Atribut yang digunakan untuk menyimpan jurusan atau program studi pengguna.
		studentIdentityNumber	String	30	Atribut yang digunakan untuk menyimpan nomor identitas pelajar atau mahasiswa.
		studentIdCardPath	String	255	Atribut yang digunakan untuk menyimpan lokasi file kartu tanda mahasiswa.
		dateOfBirth	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal lahir pengguna.
		createdAt	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal data pengguna dibuat.
		updatedAt	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal data pengguna diperbarui.
2.	Book	Id	String	36	Atribut yang digunakan sebagai ID unik buku.
		title	String	100	Atribut yang digunakan untuk menyimpan judul buku.
		imageUrl	String	255	Atribut yang digunakan untuk menyimpan URL gambar sampul buku.
		price	Int	-	Atribut yang digunakan untuk menyimpan harga buku dalam rupiah.
		stock	Int	-	Atribut yang digunakan untuk menyimpan jumlah stok buku yang tersedia.
		soldCount	Int	-	Atribut yang digunakan untuk menyimpan jumlah buku yang telah terjual.
		avarageRating	Double	-	Atribut yang digunakan untuk menyimpan nilai rata-rata rating buku.
		reviewCount	Int	-	Atribut yang digunakan untuk menyimpan jumlah ulasan pada buku.
		description	String	500	Atribut yang digunakan untuk menyimpan deskripsi buku.
3.	BookImage	id	Int	-	Atribut yang digunakan sebagai ID unik gambar buku.
		bookId	Int	-	Atribut yang digunakan untuk menyimpan ID buku yang memiliki gambar tersebut.
		imagePath	String	255	Atribut yang digunakan untuk menyimpan lokasi atau URL gambar buku.
		createdAt	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal gambar buku dibuat.
		updatedAt	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal gambar buku diperbarui.
4.	CartItem	id	String	36	Atribut yang digunakan sebagai ID unik item keranjang.
		bookId	String	36	Atribut yang digunakan untuk menyimpan ID buku yang dimasukkan ke keranjang.
		title	String	255	Atribut yang digunakan untuk menyimpan judul buku pada keranjang.
		price	Int	-	Atribut yang digunakan untuk menyimpan harga satuan buku pada keranjang.
		quantity	Int	-	Atribut yang digunakan untuk menyimpan jumlah buku yang dimasukkan ke keranjang.
		stock	Int	-	Atribut yang digunakan untuk menyimpan stok buku yang tersedia.

No.	Objek Data	Atribut Data	Tipe Data	Length	Keterangan
		weight	Int	-	Atribut yang digunakan untuk menyimpan berat buku dalam satuan gram.
		imageUrl	String	255	Atribut yang digunakan untuk menyimpan URL gambar buku pada keranjang.
5.	Order	id	String	36	Atribut yang digunakan sebagai ID unik pesanan.
		status	String	30	Atribut yang digunakan untuk menyimpan status pesanan.
		shippingAddress	String	255	Atribut yang digunakan untuk menyimpan alamat pengiriman pesanan.
		shippingCost	Int	-	Atribut yang digunakan untuk menyimpan biaya pengiriman pesanan.
		subtotal	Int	-	Atribut yang digunakan untuk menyimpan subtotal harga barang sebelum biaya tambahan atau potongan.
		totalPrice	Int	-	Atribut yang digunakan untuk menyimpan total harga keseluruhan pesanan.
		discountAmount	int	-	Atribut yang digunakan untuk menyimpan jumlah potongan diskon.
		discountName	String	100	Atribut yang digunakan untuk menyimpan nama diskon yang digunakan pada pesanan.
		discountPercentage	int	-	Atribut yang digunakan untuk menyimpan persentase diskon yang digunakan pada pesanan.
		courierService	String	20	Atribut yang digunakan untuk menyimpan nama layanan kurir pengiriman.
		courierServiceType	String	20	Atribut yang digunakan untuk menyimpan tipe layanan kurir pengiriman.
		trackingNumber	String	50	Atribut yang digunakan untuk menyimpan nomor resi pengiriman.
		paymentProof	String	255	Atribut yang digunakan untuk menyimpan URL bukti pembayaran.
		paymentDeadline	DateTime	-	Atribut yang digunakan untuk menyimpan batas waktu pembayaran pesanan.
		createdAt	DateTime	-	Atribut yang digunakan untuk menyimpan tanggal pesanan dibuat.
6.	OrderItem	id	String	36	Atribut yang digunakan untuk menyimpan daftar item buku pada pesanan.
		bookId	String	36	Atribut yang digunakan sebagai ID unik item pesanan.
		title	String	255	Atribut yang digunakan untuk menyimpan ID buku pada item pesanan.
		imageUrl	String	255	Atribut yang digunakan untuk menyimpan judul buku pada item pesanan.
		qty	int	-	Atribut yang digunakan untuk menyimpan URL gambar buku pada item pesanan.
		priceAtPurchase	int	-	Atribut yang digunakan untuk menyimpan jumlah buku pada item pesanan.
		weight	int	-	Atribut yang digunakan untuk menyimpan harga buku pada saat transaksi dilakukan.
7.	Address	id	String	36	Atribut yang digunakan untuk menyimpan berat buku pada item pesanan.
		label	String	30	Atribut yang digunakan sebagai ID unik alamat.
		recipientName	String	100	Atribut yang digunakan untuk menyimpan label alamat.

No.	Objek Data	Atribut Data	Tipe Data	Length	Keterangan
		phoneNumber	String	20	Atribut yang digunakan untuk menyimpan nama penerima.
		streetAddress	String	255	Atribut yang digunakan untuk menyimpan nomor telepon penerima.
		city	String	50	Atribut yang digunakan untuk menyimpan alamat jalan.
		cityId	String	20	Atribut yang digunakan untuk menyimpan nama kota.
		province	String	50	Atribut yang digunakan untuk menyimpan ID kota yang digunakan dalam proses perhitungan ongkos kirim.
		postalCode	String	10	Atribut yang digunakan untuk menyimpan nama provinsi.
		isPrimary	bool	-	Atribut yang digunakan untuk menyimpan kode pos.
		latitude	double	-	Atribut yang digunakan untuk menyimpan status alamat utama.
		longitude	double	-	Atribut yang digunakan untuk menyimpan koordinat lintang alamat.
8.	AppNotification	id	String	36	Atribut yang digunakan untuk menyimpan koordinat bujur alamat.
		type	String	30	Atribut yang digunakan sebagai ID unik notifikasi.
		title	String	100	Atribut yang digunakan untuk menyimpan tipe notifikasi.
		message	String	255	Atribut yang digunakan untuk menyimpan judul notifikasi.
		orderId	String	36	Atribut yang digunakan untuk menyimpan isi pesan notifikasi.
		isRead	bool	-	Atribut yang digunakan untuk menyimpan ID pesanan yang berkaitan dengan notifikasi.
		createdAt	DateTime	-	Atribut yang digunakan untuk menyimpan status notifikasi sudah dibaca atau belum.
9.	ShippingOption	id	String	36	Atribut yang digunakan untuk menyimpan tanggal notifikasi dibuat.
		name	String	50	Atribut yang digunakan sebagai ID unik opsi pengiriman.
		cost	int	-	Atribut yang digunakan untuk menyimpan nama layanan pengiriman.
		eta	String	30	Atribut yang digunakan untuk menyimpan biaya pengiriman

4.4.2 Perancangan Arsitektural

Perancangan arsitektur sistem bertujuan untuk menggambarkan bagaimana komponen-komponen sistem saling berkomunikasi dalam pengembangan perangkat lunak. Arsitektur pada Aplikasi Toko Buku *Online* tekMira mencakup sisi klien dan sisi server, dengan Flutter digunakan sebagai aplikasi *mobile* pada sisi klien. Pembangunan sisi server berada di luar lingkup tugas akhir ini, sehingga server *backend* diperlukan sebagai satu penyedia layanan yang diakses melalui REST API.

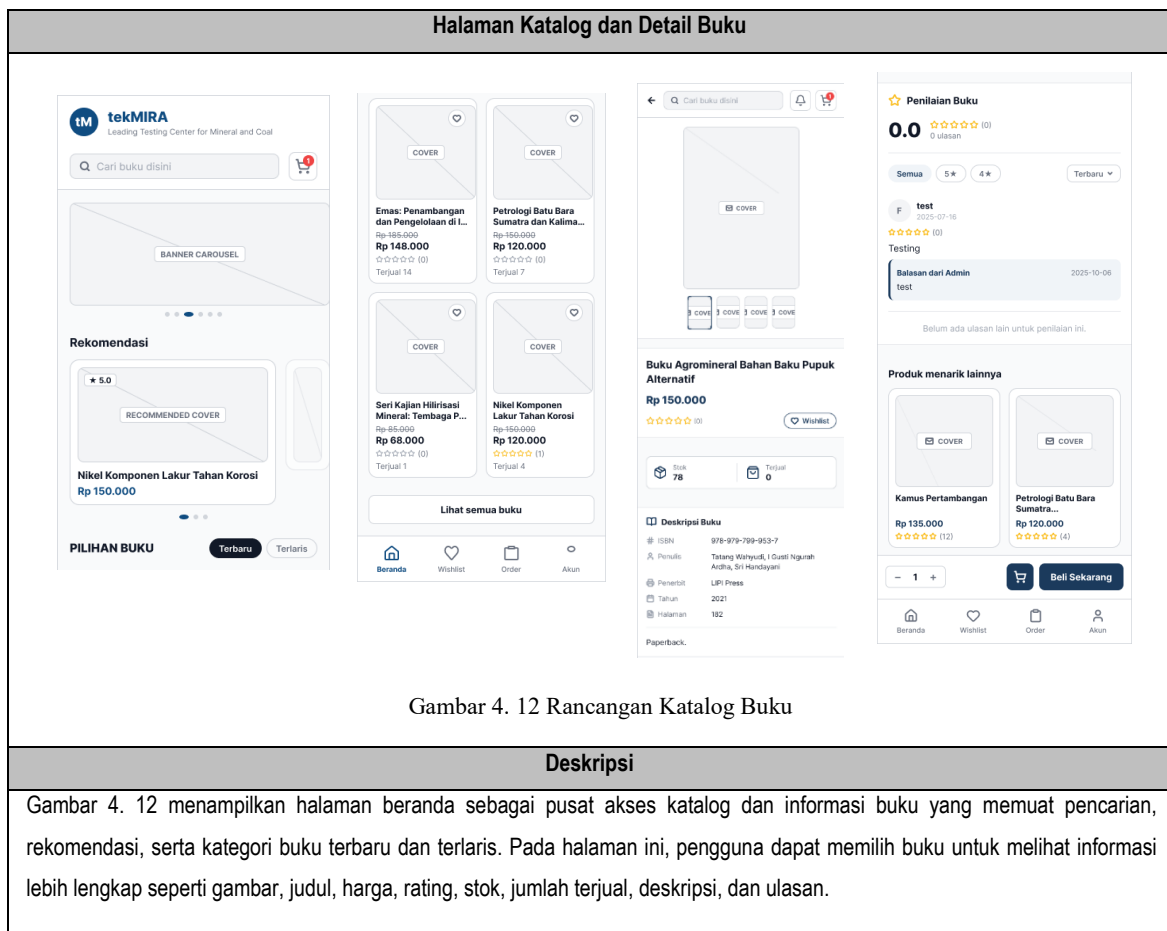


Gambar 4. 11 Perancangan Arsitektural

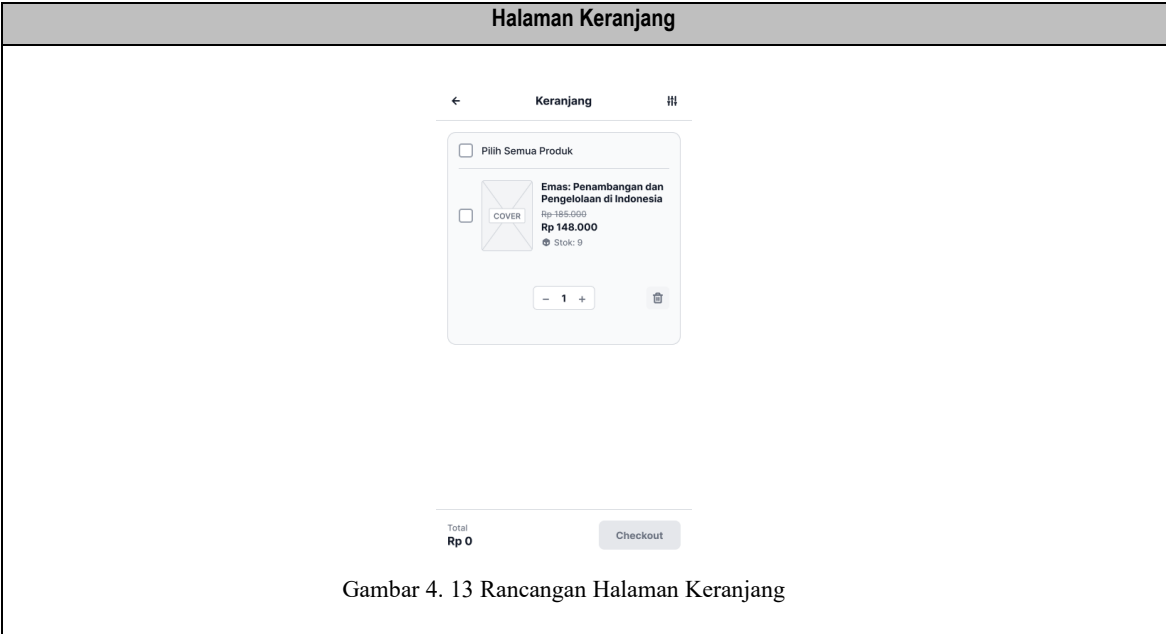
4.4.3 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan tahap yang bertujuan untuk menentukan bentuk tampilan dan susunan interaksi pada aplikasi toko buku *online* tekMira sebelum diimplementasikan ke dalam kode program. Contoh antarmuka pengguna dari aplikasi toko buku *online* tekMira dapat dilihat pada gambar

1. Halaman Katalog dan Detail Buku



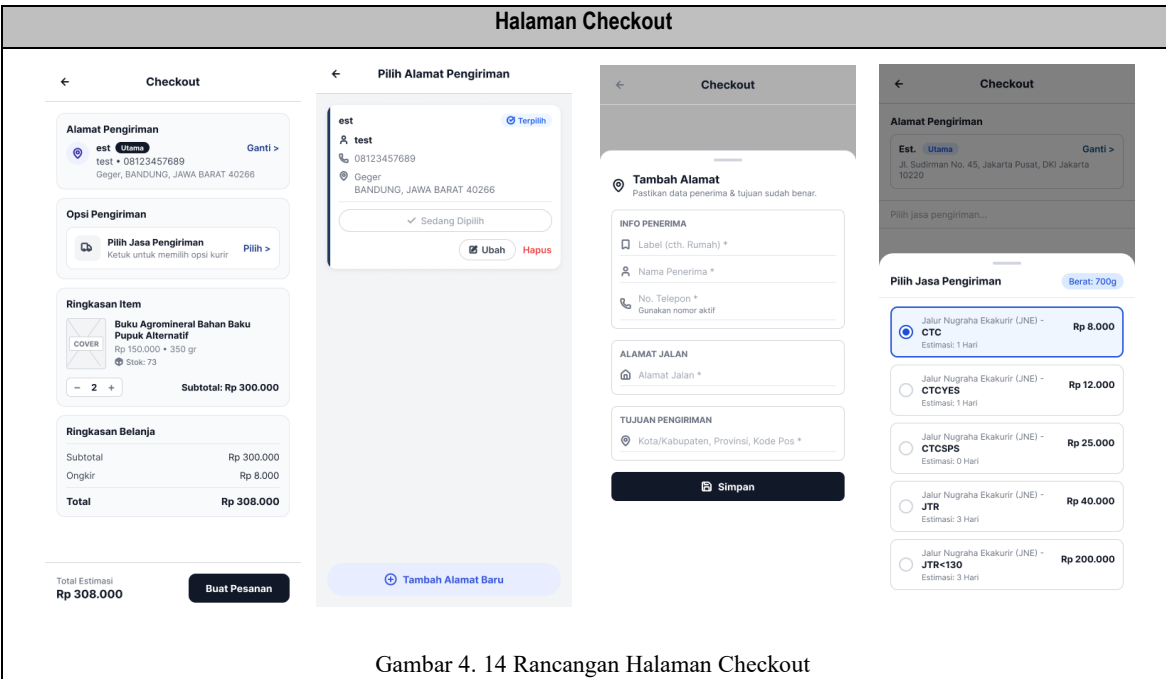
2. Halaman Keranjang



Gambar 4. 13 Rancangan Halaman Keranjang

Deskripsi
Gambar 4. 13 menampilkan rancangan halaman keranjang berisi daftar buku yang ditambahkan pengguna. Setiap item menyediakan pengaturan jumlah, penghapusan, dan pemilihan item untuk checkout.

3. Halaman Checkout



Gambar 4. 14 Rancangan Halaman Checkout

Deskripsi
Gambar 4. 14 menampilkan rancangan halaman <i>checkout</i> untuk memproses pembelian. Halaman ini menyediakan pilihan alamat, pilihan layanan pengiriman, dan ringkasan pembelian.

4. Halaman Pembayaran

Halaman Pembayaran

Pembayaran

Instruksi Pembayaran

Silakan lakukan transfer sejumlah Rp 193.000 ke rekening berikut:

Bank Mandiri

No. Rekening: 1234-5678-90

Atas Nama: BLU BBPMBT

Sbatas waktu pembayaran: 23:59:55

Unggah Bukti Pembayaran

Pilih gambar

COVER

Emas: Penambangan dan Pengolahan...
Rp 185.000

Qty 1

Gambar 4. 15 Rancangan Halaman Pembayaran

Deskripsi

Gambar 4. 15 menampilkan rancangan halaman pembayaran yang muncul setelah pesanan berhasil dibuat. Halaman ini menampilkan instruksi pembayaran, daftar item, dan total tagihan.

5. Halaman Pesanan dan Detail Pesanan

Halaman Pesanan dan Detail Pesanan

Pesanan Saya

Sedang Berlangsung

Riwayat

✓ Semua

Pending Payment

Waiting Conf...

Order ID: #106

SHIPPED

Tanggal: 10/10/2025

COVER

Emas: Penambangan dan Pengolahan di Indonesia

1 x Rp 185.000

Total: Rp 174.500

Lihat Detail

Order ID: #105

PAID

Tanggal: 10/10/2025

COVER

Emas: Penambangan dan Pengolahan di Indonesia

1 x Rp 185.000

Total: Rp 174.500

Lihat Detail

Beranda

Wishlist

Order

Akun

Detail Pesanan

#106

SHIPPED

10 Okt 2025 • 10:20

✓

Pembayaran telah dikonfirmasi. Menunggu pengiriman.

Bukti Transfer / Foto Buku

PROOF OF PAYMENT IMAGE

Nomor Resi

JNE0009

Nomor Resi

JNE0009

Alamat Pengiriman

est 081

test, BANDUNG, JAWA BARAT 40153

Rincian Produk

COVER

Emas: Penambangan d. Rp 185.000

Qty 1

Ringkasan Pembayaran

Subtotal

Rp 185.000

Diskon

- Rp 18.500

Ongkir

Rp 8.000

Total

Rp 174.500

✓ Tandai Selesai

Beranda

Wishlist

Order

Akun

Gambar 4. 16 Rancangan Halaman Pesanan

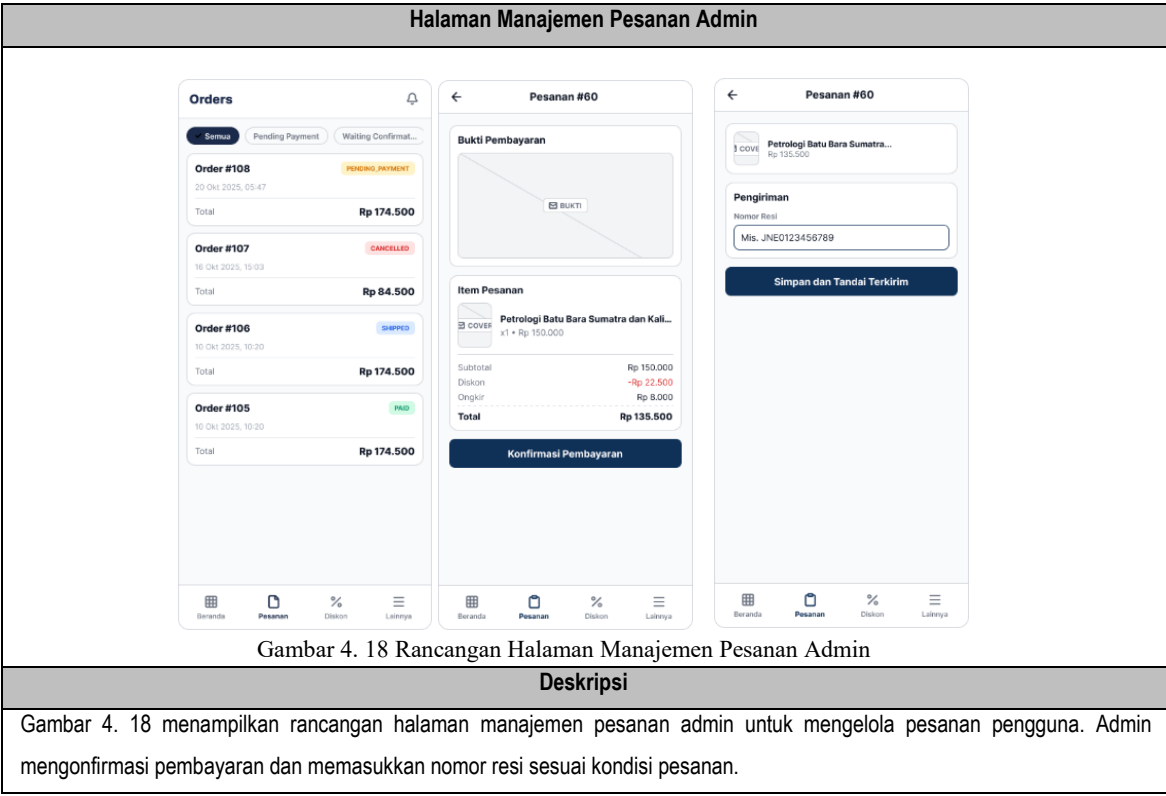
Deskripsi

Gambar 4. 16 menampilkan rancangan halaman pesanan untuk memantau status pembelian. Halaman ini terbagi atas pesanan berlangsung, yang dapat disaring berdasarkan status, dan riwayat pesanan.

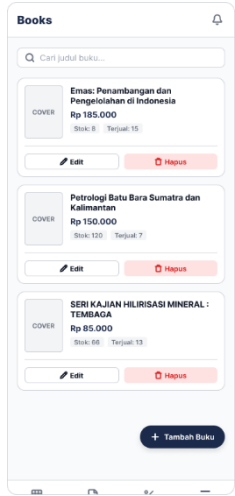
6. Halaman Dashboard Admin



7. Halaman Manajemen Pesanan Admin



8. Halaman Manajemen Buku Admin

Halaman Manajemen Buku Admin	
 Books Carilah judul buku... Emas: Penambangan dan Pengolahan di Indonesia Rp 185.000 Stok: 8 Terjual: 15 Petrologi Batu Bara Sumatra dan Kalimantan Rp 150.000 Stok: 120 Terjual: 7 SERI KAJIAN HILIRISASI MINERAL : TEMBAGA Rp 85.000 Stok: 60 Terjual: 13 + Tambah Buku	 Create Book Cover 3:4 (wajib) Pilih Gambar Unggah Gambar Reset Judul * Sistem Informasi Tambang Indonesia Deskripsi Buku referensi lengkap mengenai tata kelola hilirisasi... Harga (Rp) * 150.000 Berat (gram) 350 Stok 25 Gambar Tambahan (opsional, maks. 5) Simpan Beranda Pemasaran Diskon Laporan
Gambar 4. 19 Rancangan Halaman Buku Admin	
Deskripsi	
Gambar 4. 19 menampilkan rancangan halaman kelola buku beserta formulir tambah atau ubah data buku.	

BAB 5

IMPLEMENTASI DAN PENGUJIAN

5.1 Konstruksi Perangkat Lunak

Tahap konstruksi perangkat lunak merupakan tahap lanjutan dari perancangan yang telah dijabarkan pada Bab 4, yaitu proses penerjemahan hasil analisis kebutuhan, pemodelan data, dan perancangan kelas ke dalam bentuk kode program yang dapat dijalankan. Konstruksi Aplikasi Toko Buku *Online* tekMira dilakukan dengan memanfaatkan *framework* Flutter beserta ekosistem perangkat pendukungnya, sehingga aplikasi dapat dibangun dengan baik dan terstruktur.

5.1.1 Kebutuhan Perangkat Lunak

Kebutuhan pembangunan perangkat lunak terdiri atas dua aspek, yaitu spesifikasi minimum perangkat keras yang digunakan sebagai media pengembangan dan pengujian, serta kumpulan perangkat lunak (*tools*) yang digunakan dalam proses perancangan, pengkodean, dan pengujian aplikasi. Kedua aspek tersebut ditetapkan untuk memastikan proses pengembangan Aplikasi Toko Buku *Online* tekMira dapat berjalan tanpa kendala teknis, khususnya pada saat kompilasi kode Flutter dan penggunaan *emulator* Android.

Spesifikasi perangkat keras minimum mengacu pada persyaratan sistem resmi dari Android Studio dan Flutter SDK, mengingat kedua perangkat lunak tersebut merupakan komponen utama dalam proses pengembangan. Rincian spesifikasi tersebut disajikan pada Tabel 5.1, sedangkan daftar perangkat lunak yang digunakan selama pembangunan aplikasi disajikan pada Tabel 5.2

Tabel 5. 1 Spesifikasi Minimum Perangkat Keras Pengembangan

No.	Komponen	Persyaratan Minimum
1	Sistem Operasi	Windows 10 64-bit / macOS 12 atau lebih tinggi / Linux 64-bit
2	Processor	Intel Core i3 generasi ke-2 (atau setara) / AMD Ryzen 3
3	Ram	8 GB
4	Ruang Penyimpanan	8 GB ruang kosong (disarankan SSD)
5	Resolusi Layar	1280 x 800 piksel

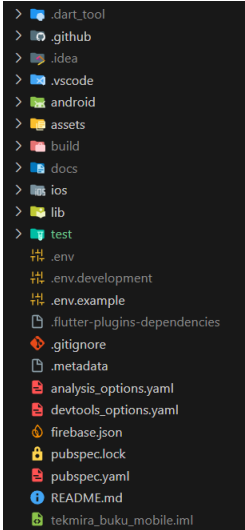
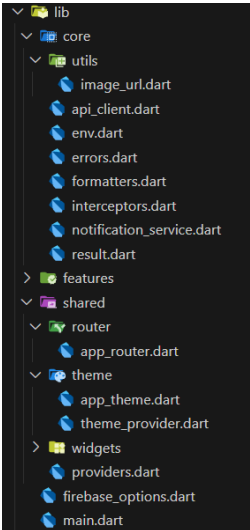
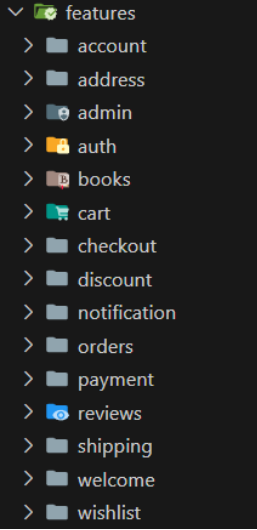
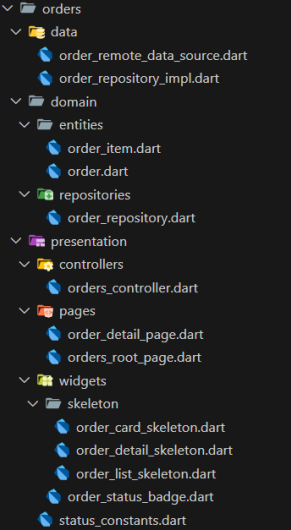
Tabel 5. 2 Perangkat Lunak yang Digunakan

No.	Nama Perangkat Lunak	Sumber	Deskripsi
1	Flutter SDK	Google LLC	<i>Framework</i> utama untuk membangun aplikasi <i>mobile cross-platform</i> berbasis Dart
2	Dart	Google LLC	Bahasa pemrograman yang digunakan dalam pengembangan aplikasi berbasis Flutter
3	Android Studio	Google LLC	<i>Integrated Development Environment</i> (IDE) resmi untuk pengembangan aplikasi Android
4	Android Emulator	Google LLC	Perangkat virtual untuk menjalankan dan menguji aplikasi tanpa perangkat fisik

5	Firebase Console	Google LLC	Konsol pengaturan <i>Firebase CLOUD Messaging (FCM)</i> untuk layanan <i>push notification</i>
6	Git	Software Conservancy	Sistem kontrol versi untuk mengelola perubahan kode program

5.2 Pengkodean

5.2.1 Tahap Pengkodean Aplikasi Mobile

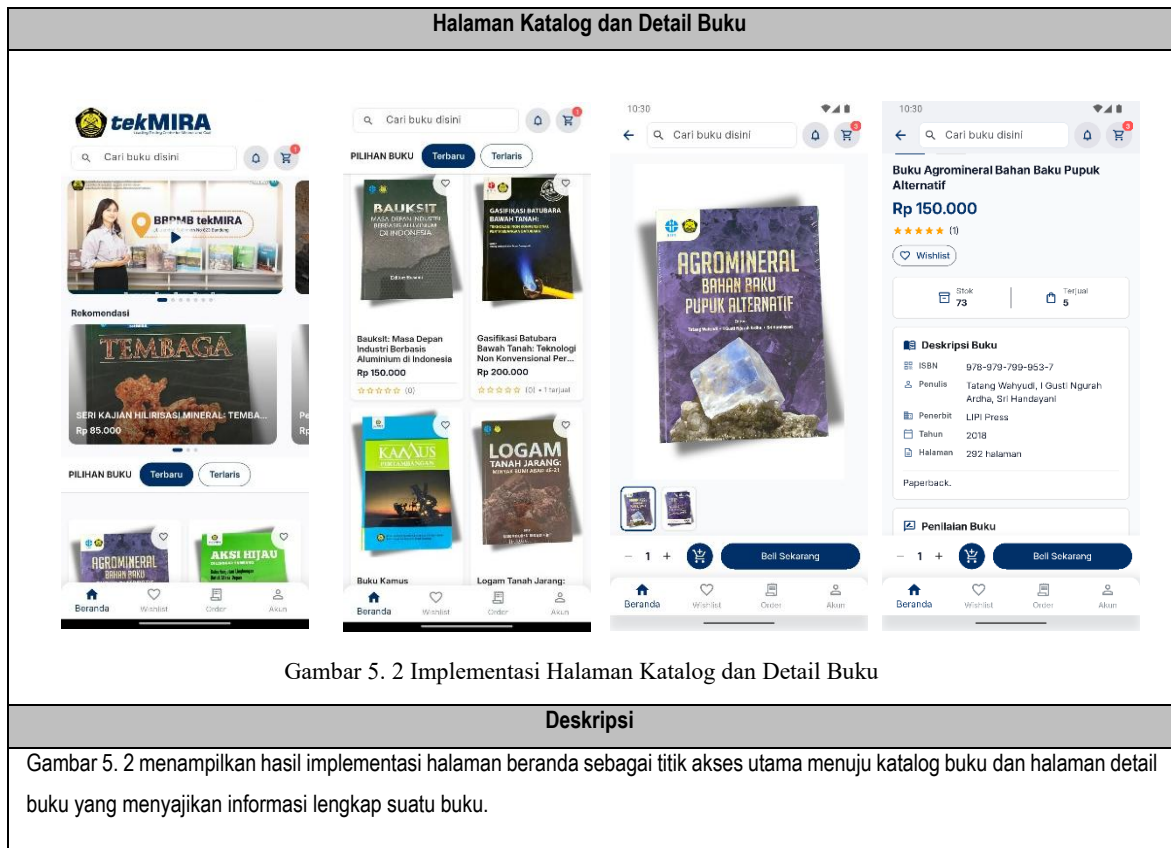
Struktur Folder Aplikasi Mobile			
			
Deskripsi			
Struktur folder aplikasi disusun menggunakan pola Feature-First Clean Architecture dengan pengelompokan kode berdasarkan tanggung jawab dan fitur. Struktur proyek secara keseluruhan menunjukkan direktori utama dan berkas pendukung aplikasi, sedangkan pada direktori lib kode sumber utama dikelompokkan ke dalam core, features, dan shared. Direktori core digunakan untuk komponen umum seperti komunikasi REST API, penanganan kesalahan, notifikasi, dan utilitas, sedangkan shared berisi komponen yang digunakan lintas fitur seperti konfigurasi navigasi, tema, widget umum, dan provider. Direktori features mengelompokkan kode berdasarkan fitur aplikasi, seperti autentikasi, katalog buku, keranjang, checkout, pesanan, pembayaran, ulasan, wishlist, dan administrasi. Setiap fitur selanjutnya dibagi menjadi lapisan data, domain, dan presentation, dengan data menangani pengelolaan sumber data, domain menangani entity dan kontrak repository, serta presentation menangani antarmuka pengguna dan pengelolaan state. Struktur tersebut diterapkan secara konsisten pada setiap fitur untuk menjaga keteraturan dan pemisahan tanggung jawab kode program.			

Gambar 5. 1 Struktur Folder Aplikasi Mobile

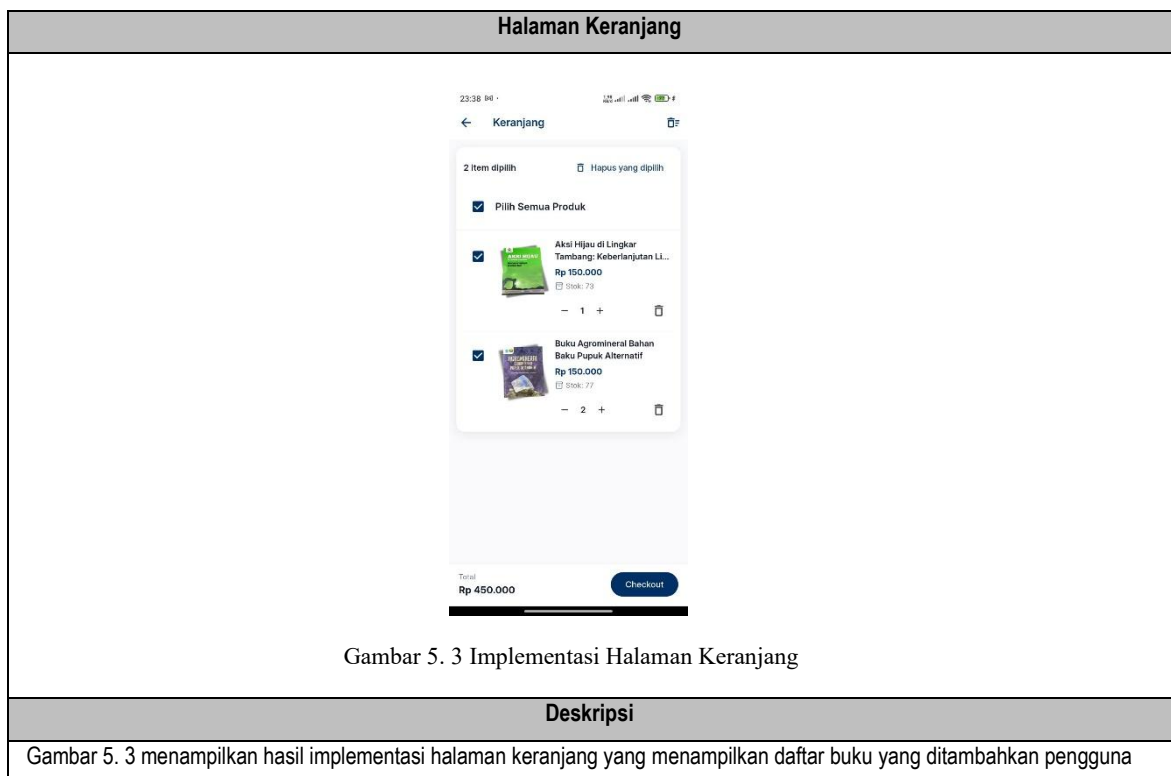
5.2.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna merupakan hasil penerapan dari perancangan antarmuka pengguna yang telah disusun sebelumnya.

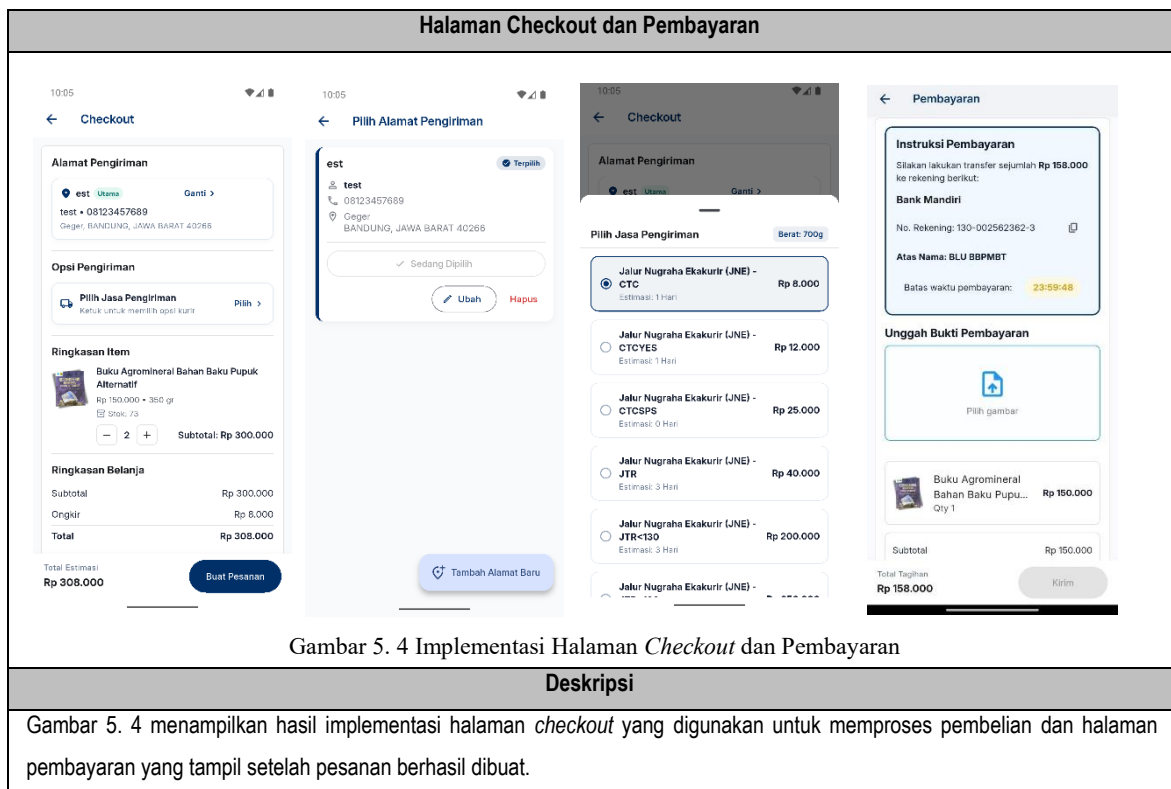
1. Halaman Katalog dan Detail Buku



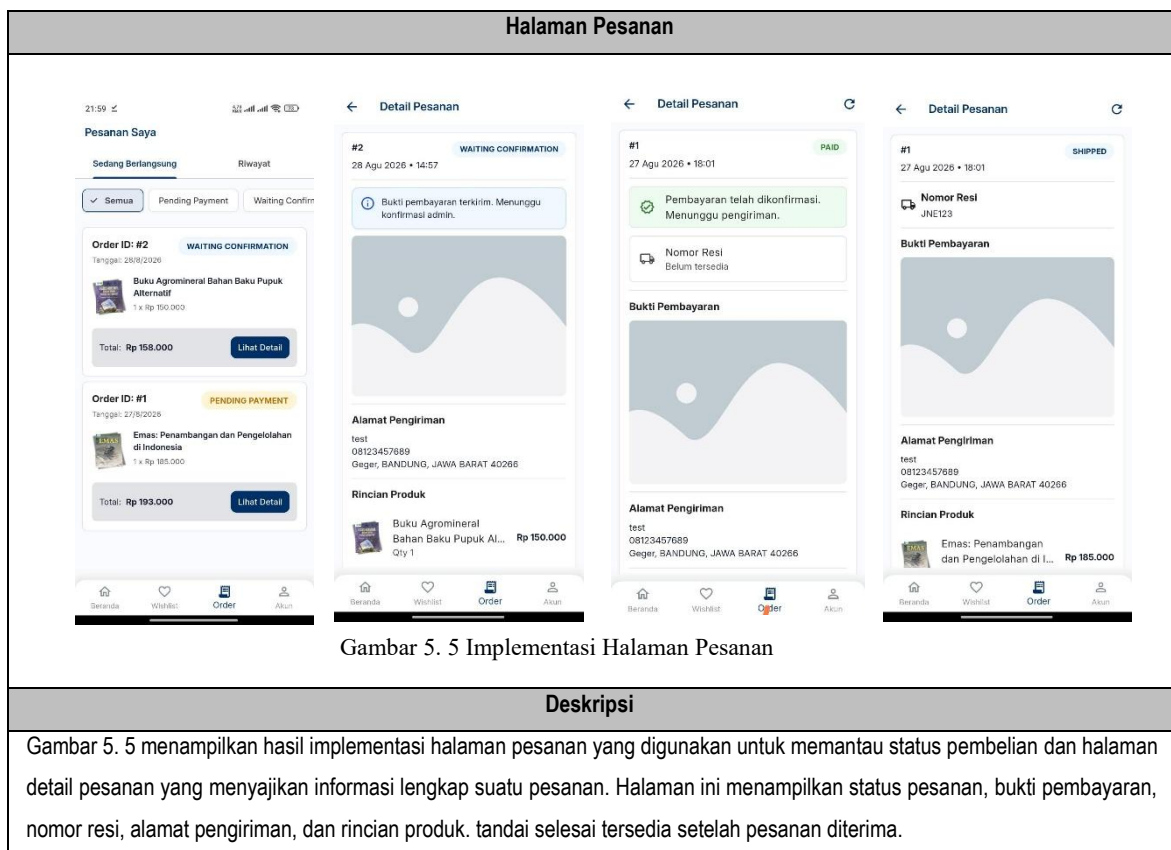
2. Halaman Keranjang



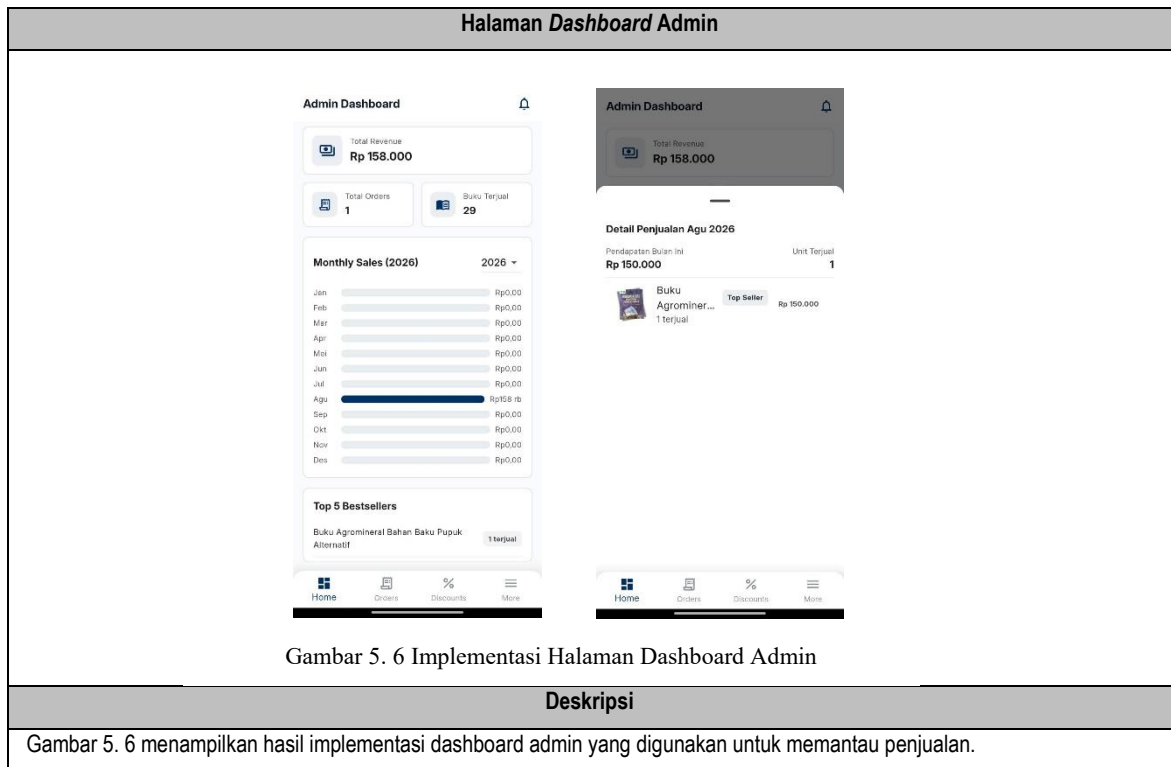
3. Halaman Checkout dan Pembayaran



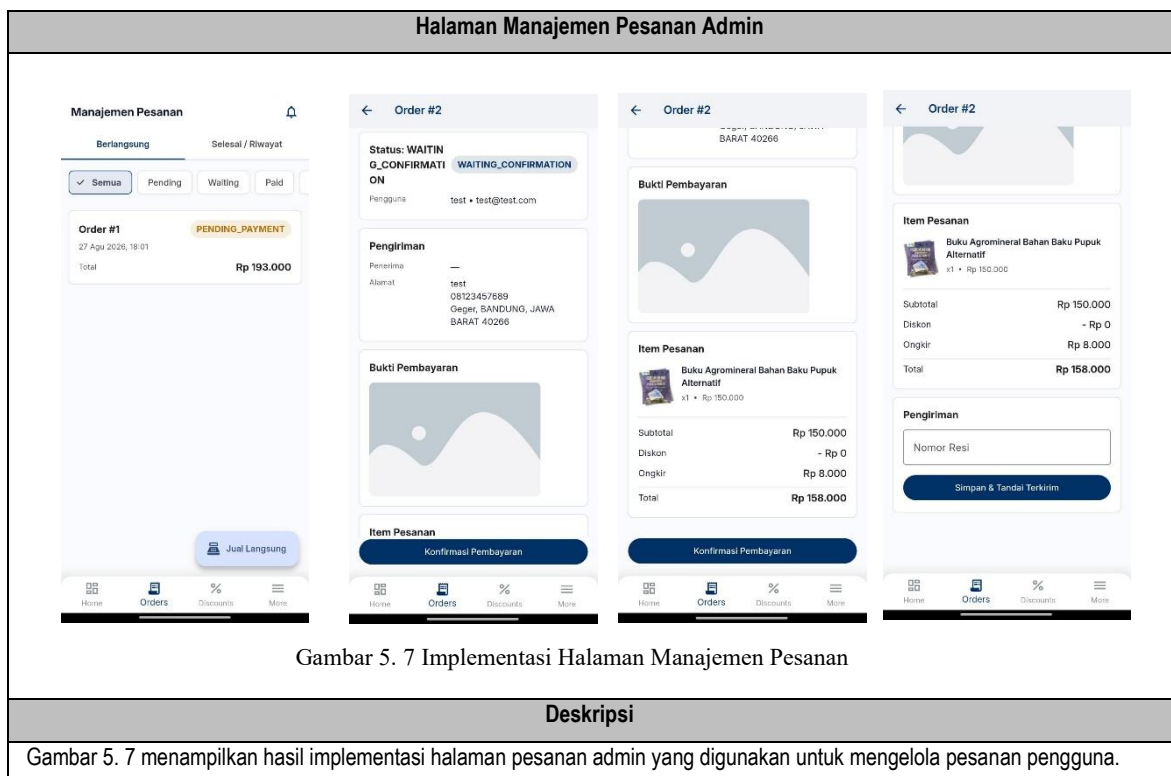
4. Halaman Pesanan dan Detail Pesanan



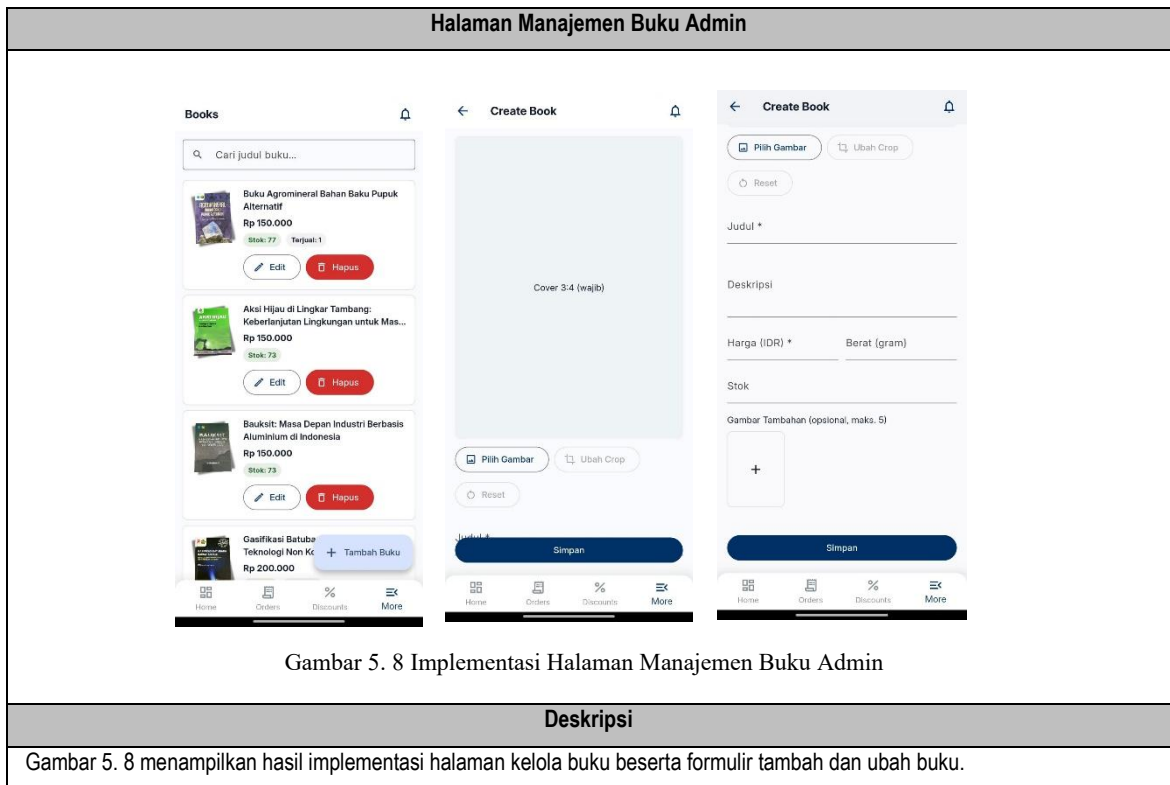
5. Halaman *Dashboard Admin*



6. Halaman Manajemen Pesanan Admin



7. Halaman Manajemen Buku Admin



5.3 Pengujian

Pengujian perangkat lunak merupakan suatu proses untuk menguji atau memastikan program yang telah dibuat sudah sesuai dengan yang diharapkan, dengan pengujian perangkat lunak ini dapat mengetahui kualitas perangkat lunak tersebut. Salah satu cara untuk melakukan pengujian perangkat lunak yaitu dengan menggunakan *black box testing*.

Pengujian ini dilakukan dengan menggunakan *real device* yaitu *smartphone* dengan sistem operasi android, antara lain : Poco M4 Pro, pengujian perangkat lunak dengan metode *black box testing* pada tugas akhir Pembangunan Aplikasi Toko Buku *Online* Berbasis Mobile Menggunakan Flutter (Studi Kasus: BBPMB tekMira) dapat dilihat pada tabel 5.2

Tabel 5. 3 Black Box Testing

Kode TS	Test Scenario	No. TC	Test Case	Test Step	Test Data	Excepted Result	Actual Result	Pass / Fail
TS-01	Membuat akun	TC-1	Membuat Akun Baru	1. Buka halaman Registrasi.. 2. Isi nama, lengkap, nama, pengguna, tanggal lahir, instansi, email, dan kata sandi 3. Tekan tombol daftar	Nama : John Doe; Username : johndoe; Tanggal Lahir : 01/01/2000 Instansi : UNPAS Email : john.doe@gmail.com; Kata sandi : JohnDoe123!	Sistem menyimpan akun baru, menampilkan pesan sukses, dan membuka halaman beranda	Akun berhasil disimpan, pesan sukses ditampilkan, dan pengguna diarahkan ke beranda	Pass
TS-02	Melihat Katalog Buku	TC-1	Menampilkan daftar buku	1. Buka halaman Beranda	-	Daftar buku tampil sesuai urutan default	Daftar buku berhasil ditampilkan sesuai urutan default.	Pass
		TC-2	Mencari dan mengurutkan buku	1. Buka halaman semua buku 2. Masukkan kata kunci 3. Pilih urutan	Kata kunci : "Emas"; Urutan : Terlaris	Hasil pencarian tampil sesuai kata kunci dan urutan	Hasil pencarian buku berhasil ditampilkan sesuai kata kunci dan urutan yang dipilih.	Pass
		TC-3	Menampilkan detail buku	1. Buka katalog. 2. Pilih buku	Buku : Nikel Komponen Lakur Tahan Korosi	Rincian buku, galeri dan ulasan tampil	Rincian buku, galeri gambar, dan ulasan berhasil ditampilkan.	Pass
TS-03	Mengelola Keranjang	TC-1	Menambah ke Keranjang	1. Buka halaman detail buku 2. Tentukan jumlah 3. Tekan icon tambah ke keranjang	Buku : Nikel Komponen Lakur Tahan Korosi; Jumlah : 2	Item tersimpan, pesan tampil	Item berhasil ditambahkan ke keranjang dan notifikasi berhasil ditampilkan.	Pass
		TC-2	Mengubah jumlah item	1. Buka halaman keranjang 2. Tekan (+)	Jumlah awal : 2	Kuantitas & total dihitung ulang	Kuantitas item dan total harga berhasil dihitung ulang sesuai perubahan jumlah.	Pass
		TC-3	Menghapus Item	1. Buka halaman keranjang 2. Tekan icon hapus	Item : Nikel Komponen Lakur Tahan Korosi	Item terhapus, daftar diperbarui	Item berhasil dihapus dari keranjang dan daftar diperbarui.	Pass

Kode TS	Test Scenario	No. TC	Test Case	Test Step	Test Data	Excepted Result	Actual Result	Pass / Fail
TS-04	Mengelola Alamat	TC-1	Menambah/memperbarui alamat	1. Buka halaman alamat 2. Klik tambah 3. Isi data 4. Cari kota 5. Tekan simpan	Penerima : John Doe; Alamat : Jl. Merdeka No.10; Kota : Bandung	Alamat tersimpan, pesan sukses	Alamat berhasil disimpan dan notifikasi keberhasilan ditampilkan.	Pass
		TC-2	Menjadikan alamat utama	1. Tekan "Jadikan utama"	Alamat : Jl. Merdeka No. 10	Status alamat utama diperbarui	Alamat berhasil ditetapkan sebagai alamat utama dan status diperbarui.	Pass
		TC-3	Menghapus Alamat	1. Tekan tombol hapus	Alamat : Jl. Merdeka No. 10	Alamat terhapus	Alamat berhasil dihapus dari daftar alamat.	Pass
TS-05	Melakukan Pemesanan	TC-1	Checkout dan membuat pesanan	1. Pilih item 2. Checkout 3. Pilih alamat 4. Pilih pengiriman 5. Buat pesanan	Item : Nikel Komponen Lakur Tahan Korosi; Layanan : JNE REG	Pesanan dibuat dengan status PENDING_PAYMENT, buka halaman pembayaran	Pesanan berhasil dibuat dengan status PENDING_PAYMENT dan pengguna diarahkan ke halaman pembayaran.	Pass
		TC-2	Mengunggah Bukti Pembayaran	1. Buka pembayaran 2. Pilih file 3. Unggah	Order ID: #1; File : bukti_transfer_johndoe.jpg	Status berubah ke WAITING_CONFIRMATION.	Bukti pembayaran berhasil diunggah dan status pesanan berubah menjadi WAITING_CONFIRMATION.	Pass
TS-06	Melihat Riwayat Pesanan	TC-1	Melihat daftar dan Detail Pesanan	1. Buka halaman Order 2. Tekan "Lihat Detail"	Order ID: #1	Daftar dan rincian pesanan tampil	Daftar pesanan dan rincian pesanan berhasil ditampilkan sesuai status yang dipilih.	Pass
		TC-2	Menerima notifikasi perubahan status pesanan	1. Buka halaman notifikasi 2. Pilih notifikasi pesanan	Notifikasi : "Pembayaran Pesanan #1 dikonfirmasi"	Notifikasi perubahan status pesanan dan pembayaran ditampilkan kepada pengguna	Notifikasi perubahan status pesanan dan pembayaran berhasil ditampilkan kepada pengguna	Pass
TS-14	Melihat Notifikasi	TC-1	Menampilkan daftar notifikasi	1. Buka notifikasi	Pengguna: John Doe	Daftar pemberitahuan tampil	Daftar pemberitahuan berhasil ditampilkan.	Pass
		TC-2	Menandai satu notifikasi dibaca	1. Tekan salah satu notifikasi	Notifikasi : "Pembayaran Pesanan #1 dikonfirmasi"	Status jadi terbaca	Notifikasi yang dipilih berhasil ditandai sebagai telah dibaca.	Pass
		TC-3	Menandai semua notifikasi dibaca	1. Tekan "Tandai semua dibaca"	Pengguna : John Doe	Seluruh notifikasi terbaca	Seluruh notifikasi berhasil ditandai sebagai telah dibaca.	Pass

Kode TS	Test Scenario	No. TC	Test Case	Test Step	Test Data	Excepted Result	Actual Result	Pass / Fail
TS-07	Melihat Dashboard Admin	TC-1	Menampilkan KPI, grafik dan top 5 buku	1. Buka dashboard 2. Pilih tahun	Admin: tekmira; Tahun : 2026	KPI, grafik bulanan, top 5 buku tampil	KPI, grafik penjualan bulanan, dan lima buku teratas berhasil ditampilkan sesuai tahun yang dipilih.	Pass
		TC-2	Melihat rincian penjualan bulanan	1. Tekan batang grafik bulan	Bulan : Agustus 2026	Rincian penjualan bulanan tampil	Rincian omzet untuk bulan yang dipilih berhasil ditampilkan.	Pass
TS-08	Mengelola pesanan admin	TC-1	Menampilkan daftar dan detail pesanan	1. Filter status 2. Tekan salah satu pesanan	Status: Menunggu Konfirmasi	Daftar dan rincian pesanan tampil	Daftar dan rincian pesanan berhasil ditampilkan sesuai status yang dipilih.	Pass
		TC-2	Mengonfirmasi pembayaran	1. Periksa bukti pembayaran 2. Tekan konfirmasi pembayaran 3. Konfirmasi	Pesanan : Order ID #1	Status pesanan menjadi Paid	Bukti pembayaran berhasil dikonfirmasi dan status pesanan berubah menjadi paid.	Pass
		TC-3	Menginput nomor resi	1. Masukkan resi 2. Kirim pesanan	Resi : JNE001234567	Status jadi Shipped	Nomor resi berhasil disimpan dan status pesanan berubah menjadi shipped.	Pass
		TC-4	Mencatat transaksi tunai	1. Tekan "Jual Langsung" 2. Isi Data Pembeli 3. Tambah buku 4. Simpan transaksi	Pelanggan : John Smith Tanggal : 20 Juli 2026 Metode : Tunai Buku : Nikel Komponen Lakur Tahan Korosi	Pesanan dibuat, stok berkurang	Transaksi penjualan tunai berhasil disimpan sebagai pesanan dan stok buku berkurang.	Pass
		TC-5	Menerima notifikasi pesanan yang memerlukan konfirmasi pembayaran	1. Membuat pesanan dari sisi <i>customer</i> 2. Mengunggah bukti pembayaran 3. Buka halaman notifikasi admin	Order ID : #1 Status : Waiting_confirmation	Admin menerima dan dapat melihat notifikasi bahwa terdapat pesanan yang memerlukan konfirmasi pembayaran.	Admin berhasil menerima dan melihat notifikasi pesanan yang memerlukan konfirmasi pembayaran.	Pass

Kode TS	Test Scenario	No. TC	Test Case	Test Step	Test Data	Excepted Result	Actual Result	Pass / Fail
TS-09	Mengelola Buku Admin	TC-1	Menampilkan daftar buku	1. Buka halaman buku 2. Masukkan kata kunci pencarian	Kata kunci : "Emas"	Daftar buku tampil dan buku sesuai kata kunci tampil	Daftar buku berhasil ditampilkan sesuai kata kunci yang dimasukkan.	Pass
		TC-2	Menambah buku baru	1. Tekan tombol tambah 2. Isi Data 3. Unggah sampul 4. Simpan	Judul : Nikel; Harga: 95000; Stok: 20; Cover: buku_nikel.jpg	Buku baru tersimpan	Data buku baru beserta sampul dan galeri berhasil disimpan.	Pass
		TC-3	Mengubah data buku	1. Tekan edit buku 2. Ubah data/stok/foto	Buku : Nikel Komponen Lakur Tahan Korosi; Stok baru : 10	Perubahan tersimpan	Perubahan data, stok, dan foto buku berhasil disimpan.	Pass
		TC-4	Menghapus buku	1. Tekan tombol hapus	Buku: Nikel	Buku terhapus	Data buku berhasil dihapus setelah konfirmasi.	Pass

BAB 6

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pembangunan yang telah dilakukan, Aplikasi Toko Buku Online tekMIRA berhasil dibangun sebagai aplikasi *mobile* menggunakan Flutter yang mendukung proses penjualan, pencatatan transaksi, dan pelaporan penjualan secara terpusat. Aplikasi menyediakan fungsi bagi *customer* untuk membuat akun, melihat katalog buku, mengelola keranjang dan alamat pengiriman, melakukan pemesanan, serta melihat riwayat dan detail pesanan. Sementara itu, pada sisi admin tersedia fungsi untuk melihat *dashboard* penjualan, mengelola pesanan, dan mengelola data buku. Fungsi-fungsi tersebut telah disesuaikan dengan kebutuhan fungsional dan *Use Case* yang telah ditetapkan.

Aplikasi *mobile* diimplementasikan menggunakan Flutter dan terintegrasi dengan sistem *backend* melalui layanan REST API untuk mendukung pertukaran data pada fungsi-fungsi yang tersedia. Proses pembangunan dilakukan secara sistematis melalui tahapan identifikasi masalah, pengumpulan data, *planning*, *design*, *develop*, dan *testing*. Berdasarkan hasil pengujian fungsional, fungsi yang telah diimplementasikan dapat berjalan sesuai dengan kebutuhan yang ditetapkan sehingga aplikasi dapat digunakan sebagai sarana pendukung proses penjualan publikasi tekMIRA secara lebih terpusat.

6.2 Saran

Berdasarkan hasil pembangunan dan pengujian Aplikasi Toko Buku Online tekMIRA, terdapat beberapa saran yang dapat menjadi pertimbangan untuk pengembangan selanjutnya, yaitu:

- 1 Pengembangan sistem pembayaran dapat dilakukan dengan mengintegrasikan *payment gateway* agar proses pembayaran dapat dilakukan secara otomatis dan status pembayaran dapat diperbarui secara langsung pada sistem.
- 2 Pengembangan fitur pengelolaan dan pelaporan penjualan dapat dilakukan secara lebih lanjut dengan menyediakan informasi dan laporan penjualan yang lebih lengkap dan terstruktur sehingga dapat mendukung pemantauan transaksi, penjualan, serta pengelolaan publikasi oleh pihak tekMIRA.
- 3 Pelaksanaan *User Acceptance Testing* (UAT) dapat dilakukan dengan melibatkan pengguna secara langsung untuk memastikan aplikasi telah sesuai dengan kebutuhan dan proses operasional di tekMIRA sebelum aplikasi dirilis.
- 4 Rilis dan pemeliharaan aplikasi dapat dilakukan setelah aplikasi dinyatakan sesuai berdasarkan hasil UAT. Pemeliharaan secara berkala diperlukan untuk menangani kendala yang ditemukan setelah aplikasi digunakan serta menyesuaikan aplikasi dengan kebutuhan operasional yang berkembang.

DAFTAR PUSTAKA

- [ABA24] Abadi, Adelano Nur, Ardiani, Farida, “PERANCANGAN SISTEM INFORMASI MANAJEMEN STOK DAN PENJUALAN BUKU BERBASIS ANDROID PADA MUSI BOOKSTORE,” Jurnal Teknologi Informasi, Jilid 5, Nomor 3, Desember 2024.
- [BRE22] Bregaster, Sagita Rusdianto, Denny, Santoso, Edy, “Pengembangan Aplikasi Toko Buku berbasis Web Studi Kasus Toko Buku Online Yukmoco,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Jilid 6, Desember 2022.
- [FCM26] Google LLC, “Firebase Cloud Messaging,” tersedia : Mei 2026, <https://firebase.google.com/docs/cloud-messaging>, Mei 2026.
- [GÜL21] Gülcüoğlu, Ekrem, Üstün, Ahmet Berk, Seyhan, Neşet, “Comparison of Flutter and React Native Platforms,” Internet Applications and Management, Desember 2021.
- [HAR25] Hartati, Tuti, Azwar, L M, Fayyadh, Al, “Perancangan Aplikasi Penjualan Buku Berbasis Android pada CV Kamalia Sinar Sejahtera,” Jurnal ICT : Information Communication & Technology, Jilid 25, Nomor 1, 2025.
- [LIY22] Liyunjira, I Putu Dody Lesmana, “Design and Build a Mobile Bookstore Application in Order to Attract Consumers’ Reading Interest Using Prototyping Method,” Journal of Disruptive Learning Innovation (JODLI), Jilid 3, Nomor 2, 2022.
- [MOR17] Moroney, Laurence, “The definitive guide to firebase: Build Android apps on google’s mobile platform,” Apress Media LLC, Seattle, 2017.
- [PRA25] Prasetya, Rizki Nanda Winata, Muhammad Sukron, Rizki Hidayat Pane, Muhammad Hafiz Al Fazri, Muhammad Alda, “Sistem Informasi Perencanaan Aplikasi Penjualan Buku Online Berbasis Android,” Bulletin of Computer Science Research, Jilid 5, Nomor 2, Februari 2025.
- [PRE10] Roger S. Pressman, “Software Engineering: A Practitioner’s Approach,” Edisi 7, McGraw-Hill Higher Education, New York, 2010.
- [TUR15] Turban, Efraim, King, David, Lee, Jae Kyu, Liang, Ting-Peng, Turban, Deborrah C, “Electronic Commerce: A Managerial and Social Networks Perspective,” Edisi 8, Springer, Cham, 2015.
- [WAS10] Wasserman, Anthony I., “Software Engineering Issues for Mobile Application Development,” Proceedings of the FSE/SDP workshop on Future of Software Engineering Research (FoSER 2010), November 2010.
- [WIN20] Eric Windmill, “Flutter in Action,” Manning, New York, 2020.

LAMPIRAN A

BERITA ACARA WAWANCARA



Program Studi Teknik Informatika
Universitas Pasundan Bandung

BERITA ACARA WAWANCARA

Dengan ini menerangkan bahwa mahasiswa yang bertanda tangan di bawah ini:

Nama : Herdy Sya'banul Hakim
NRP : 223040164
Prodi : Teknik Informatika

Telah melakukan wawancara dengan :

Nama : Deni Nurul kamal
Jabatan : Pranata Komputer Penyelia
Instansi / : BBPMB tekMIRA
Perusahaan

Dalam rangka penyusunan Tugas Akhir tahun akademik 2025/2026 yang berjudul "PEMBANGUNAN APLIKASI TOKO BUKU ONLINE BERBASIS MOBILE MENGGUNAKAN FLUTTER (STUDI KASUS : Balai Besar Pengujian Mineral dan Batubara tekMIRA)"

Pewawancara

Herdy Sya'banul Hakim
NRP. 22.304.0164

Bandung, 24 April 2026

Narasumber



Gambar A. 1 Berita Acara Wawancara dengan Pihak tekMIRA



DAFTAR PERTANYAAN WAWANCARA

1. Bagaimana mekanisme penjualan buku sebelumnya ?

Penjualan buku produksi tekMIRA sebelumnya dilakukan melalui open booth di : pameran-pameran, acara pertemuan yang diselenggarakan oleh ditjen minerba, melalui media sosial dan market place

2. Siapa saja yang terlibat dalam proses tersebut ?

Proses penjualan buku dilakukan oleh Subkoordinator Huku , kerja sama dan informasi sebagai vocal point penjualan buku, dibantu oleh Subkoordinator Keuangan melalui bendahara penerimaan untuk pencatatan penjualan

3. Jenis publikasi apa saja yang dijual ?

Buku yang dijual adalah buku seputar teknologi pertambangan, pengolahan dan pemanfaatan mineral dan Batubara diindonesia yang merupakan hasil tulisan dari para ahli yang dimiliki oleh tekMIRA

4. Berapa rata-rata transaksi penjualan publikasi per bulan atau per tahun ?

Tahun 2025 buku yang berhasil dijual sejumlah 122 Eksmp dengan rata-rata penjualan 10 buku per bulan

5. Bagaimana mekanisme pencatatan transaksi sebelumnya ?

Pencatatan transaksi sebelumnya dilakukan secara manual oleh bendahara penerimaan, kemudian data disinkronkan dengan subkoordinator Hukum, Kerjasama dan Informasi (HKI)

6. Apakah ada kendala dari proses sebelumnya ?

Kendala yang sering terjadi adalah ketidakcocokan data penjualan di HKI dengan Bendahara, karena pembayaran yang dilakukan oleh pembeli ada yang melalui transfer ada yang langsung cash disaat pameran, karena pencatatan manual , update data penjualan kadang lama,

Gambar A. 2 Daftar Pertanyaan Wawancara Bagian 1



7. Fitur apa saja yang diharapkan ada dalam aplikasi ?

Aplikasi dapat bertindak juga sebagai marketplace resmi penjualan buku tekMIRA, sehingga semua pencatatan penjualan dilakukan di dalam satu aplikasi, disediakan fitur untuk admin menambahkan transaksi penjualan apabila penjualan dilakukan secara langsung (cash) ke pelanggan tidak melalui transfer, sehingga semua proses penjualan dapat langsung tercatat di dalam aplikasi, memudahkan proses rekap dan pelaporan penjualan kepada manajemen menjadi lebih cepat, kalau bisa ada dashboard khusus untuk manajemen untuk memantau proses penjualan secara real time

8. Mengapa dibutuhkan sistem aplikasi sendiri tidak di e-commerce lain ?

Aplikasi dapat bertindak juga sebagai marketplace resmi penjualan buku tekMIRA, sehingga semua pencatatan penjualan dilakukan didalam satu aplikasi, untuk memudahkan monitoring dan pencatatan, jadi hanya ada satu platfom yang digunakan , selain itu untuk memudahkan proses administrasi keuangan karena BIU tekMIRA adalah instansi pemerintah yang harus memenuhi aturan yang berlaku dalam hal administrasi keuangan dan publikasi instansi