

**RANCANG BANGUN *BACK-END E-COMMERCE* U-MARKET  
BERBASIS *WEBSITE* MENGGUNAKAN *FRAMEWORK LARAVEL*  
DENGAN METODE *SCRUM***

**(Skripsi)**

**Oleh**

**ANNISA TRI HAPSARI**

**2215061025**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2026**

**RANCANG BANGUN *BACK-END E-COMMERCE* U-MARKET  
BERBASIS *WEBSITE* MENGGUNAKAN *FRAMEWORK LARAVEL*  
DENGAN METODE *SCRUM***

**Oleh**

**ANNISA TRI HAPSARI**

**Skripsi**

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

**Pada**

**Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2026**

## ABSTRAK

### RANCANG BANGUN *BACK-END E-COMMERCE* U-MARKET BERBASIS *WEBSITE* MENGGUNAKAN *FRAMEWORK LARAVEL* DENGAN METODE *SCRUM*

Oleh

ANNISA TRI HAPSARI

Penelitian ini berfokus pada pengembangan sisi *back-end* yang mencakup pengelolaan data pengguna, produk, dan transaksi melalui REST API. Metode *Scrum* diterapkan untuk mendukung pengembangan sistem secara iteratif melalui 4 tahap *sprint*, sehingga setiap fitur dapat dikembangkan, diuji, dan dievaluasi secara bertahap. Pengujian sistem dilakukan menggunakan metode *black box testing* untuk memastikan seluruh fungsi berjalan sesuai kebutuhan, serta *stress testing* menggunakan Apache JMeter untuk mengukur performa sistem dalam menangani beban pengguna. Hasil pengujian menunjukkan bahwa seluruh *endpoint* API yang dikembangkan berjalan dengan baik dan valid sesuai skenario pengujian. Selain itu, berdasarkan hasil *stress testing* dengan variasi beban hingga 300 sampel pengguna, sistem mampu memberikan respon yang stabil tanpa mengalami kegagalan signifikan, sehingga menunjukkan bahwa performa *back-end* cukup andal dalam menangani permintaan secara bersamaan. Implementasi fitur utama seperti autentikasi, manajemen produk, keranjang, transaksi, serta integrasi *payment gateway* Midtrans juga berjalan sesuai fungsinya. Dengan demikian, sistem U-Market mampu menjadi solusi yang lebih terstruktur, aman, dan efisien dibandingkan metode sebelumnya, serta penerapan metode *Scrum* terbukti meningkatkan efektivitas proses pengembangan.

Kata kunci: *e-commerce*, *back-end*, Laravel, *Scrum*, REST API, U-Market

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT OF THE U-MARKET E-COMMERCE BACK-END WEB SYSTEM USING THE LARAVEL FRAMEWORK WITH THE SCRUM METHOD**

**By**

**ANNISA TRI HAPSARI**

This research focuses on back-end development, including the management of user, product, and transaction data through a structured REST API. The Scrum method is applied to support iterative and incremental system development through 4 stages of sprints, allowing each feature to be developed, tested, and evaluated progressively. System testing is conducted using black box testing to ensure all functionalities operate as intended, and stress testing using Apache JMeter to evaluate system performance under concurrent user loads. The testing results indicate that all developed API endpoints function correctly and meet the defined testing scenarios. Furthermore, stress testing with varying loads of up to 300 samples demonstrates that the system maintains stable performance without significant failures, indicating that the back-end is capable of handling concurrent requests effectively. The implementation of core features, including user authentication, product management, shopping cart, transaction processing, and integration with the Midtrans payment gateway, operates as expected. Therefore, the U-Market system provides a more structured, secure, and efficient solution compared to the previous method, while the application of the Scrum method contributes to improving the effectiveness of the development process and the quality of the resulting system.

**Keyword:** e-commerce, back-end, Laravel, Scrum, REST API, U-Market.

Judul Skripsi

: **RANCANG BANGUN *BACK-END*  
E-COMMERCE U-MARKET BERBASIS  
WEBSITE MENGGUNAKAN *FRAMEWORK*  
LARAVEL DENGAN METODE *SCRUM***

Nama Mahasiswa

: **Annisa Tri Hapsari**

Nomor Pokok

: **2215061025**

Mahasiswa

Program Studi

: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

: **Teknik**

**MENYETUJUI**

1. **Komisi Pembimbing**

Pembimbing Utama

Pembimbing Pendamping

**Yessi Mulyani, S.T., M.T.**  
NIP. 19731226 200012 2 001

**Nurrahma, M.T.**  
NIP. 19971106 202406 2 001

2. **Mengetahui**

Ketua Jurusan  
Teknik Elektro

Ketua Program Studi  
Teknik Informatika

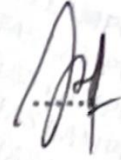
**Herlinawati, S.T., M.T.**  
NIP. 19710314 199903 2 001

**Yessi Mulyani, S.T., M.T.**  
NIP. 19731226 200012 2 001

**MENGESAHKAN**

1. Tim Penguji

Pembimbing 1 : **Yessi Mulyani, S.T., M. T.**

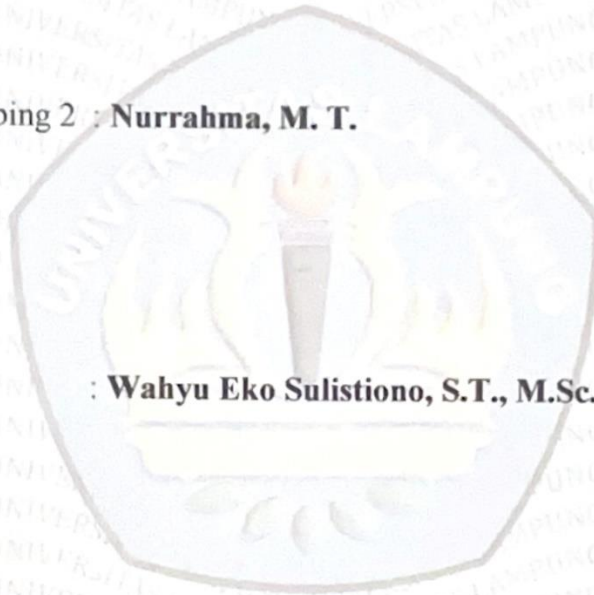
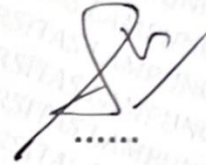


Pembimbing 2 : **Nurrahma, M. T.**



Penguji

: **Wahyu Eko Sulistiono, S.T., M.Sc.**



2. Dekan Fakultas Teknik Universitas Lampung

**Dr. H. Ahmad Herison, S.T., M. T.**

NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi: **11 Mei 2026**

## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Rancang Bangun *Back-End E-Commerce U-Market Berbasis Website Menggunakan Framework Laravel Dengan Metode Scrum*” dibuat oleh saya sendiri. Semua hasil yang tertuang pada skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila pernyataan saya tidak benar dan terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 11 Mei 2026

Penulis,



Annisa Tri Hapsari

NPM. 2215061025

## RIWAYAT HIDUP



Penulis dilahirkan di Pringsewu, pada tanggal 3 September 2004. Penulis merupakan putri bungsu dari tiga bersaudara, anak dari pasangan Bapak Sigit Iswanto dan Ibu Yuliarni. Penulis menyelesaikan pendidikannya di TK Baitussalam Pringsewu Selatan pada tahun 2010, MI Daarul Huffazh pada tahun 2016, SMP Al-Kautsar Bandar Lampung pada tahun 2019, dan SMA Negeri 1 Pringsewu pada tahun 2022.

Pada tahun 2022 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SNMPTN. Selama masa studi, penulis memfokuskan untuk mendalami bidang konsentrasi Rekayasa Perangkat Lunak. Pada tahun 2024, penulis mengikuti kegiatan Studi Independen di PT Stechoq Robotika Indonesia dengan *course* UI/UX *Specialist*. Pada tahun 2025, penulis melaksanakan Kerja Praktik di PT KAI Divre IV Tanjungkarang di Unit Hubungan Masyarakat. Selain itu, penulis juga aktif sebagai anggota Departemen Kaderisasi dan Pengembangan Organisasi Himpunan Mahasiswa Teknik Elektro selama dua periode. Di luar kegiatan akademik, penulis merupakan anggota Ikatan Muli Mekhanai Kabupaten Pringsewu sebagai RU 2 Muli Pringsewu 2024.

**PERSEMBAHAN**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

**Alhamdulillah, Atas Berkat Rahmat Allah yang Maha Kuasa**

**KUPERSEMBAHKAN KARYA INI UNTUK**

*Papah dan Mamah Tercinta*

**Sigit Iswantoro dan Yuliarni**

*Kedua Uniku*

**Aulia Puspita Rani dan Shafira Bella Sukma**

*Keponakanku*

**Hazeera Ayura**

Keluarga Besar yang selalu memberi semangat,  
Dosen yang telah membimbing dengan tulus,  
serta seluruh teman yang setia membersamai setiap langkah perjalanan ini

## **MOTTO**

“Maka barang siapa mengerjakan kebaikan sekecil apapun, niscaya dia akan melihat balasan-Nya.”

**(QS. Al-Zalzalah: 7)**

*“It’s fine to fake it  
Until you make it  
Until you do  
Until it’s true.”*

**(Taylor Swift)**

*“Take every chance and drop every fear.”*

**(Penulis)**

## SANWACANA

Segala puji syukur kehadirat Allah SWT, Tuhan semesta alam yang Maha Pengasih lagi Maha Penyayang, atas limpahan rahmat, taufik serta hidayah-Nya sehingga penulis dapat menyelesaikan penulisan skripsi dengan judul **“Rancang Bangun Back-End E-Commerce U-Market Berbasis Website Menggunakan Framework Laravel Dengan Metode Scrum”**. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung. Penulisan skripsi ini tidak lepas dari bantuan, bimbingan, motivasi, dan saran yang diberikan dari semua pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Keluarga penulis selamanya, Almarhum Bapak Sigit Iswanto, Ibu Yuliarni, Uni Aulia Puspita Rani, Uni Shafira Bella Sukma, Kak Yugo Verdinan, Kak Prabowo Pangestu, dan Hazeera Ayura yang menjadi motivasi terbesar bagi penulis untuk bertahan hidup, serta senantiasa mengajarkan kebaikan, memberikan perhatian, semangat, materi, kasih sayang, do'a terbaik, dukungan, serta kepercayaan atas setiap keputusan yang diambil oleh penulis hingga detik ini.
2. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan Dosen Pembimbing Utama yang telah memberikan bimbingan, perhatian dan dukungan dalam pengerjaan skripsi ini.
3. Ibu Nurrahma, M.T. selaku Dosen Pembimbing Pendamping yang telah memberikan bimbingan, perhatian dan dukungan dalam pengerjaan skripsi ini.
4. Bapak Wahyu Eko Sulistiono, S.T., M.sc. selaku Dosen Penguji yang telah memberikan masukan dan saran yang membangun dalam penyelesaian penulisan skripsi ini.
5. Bapak Dr. Hi. Ahmad Herison, S.T., M. T. selaku Dekan Fakultas Teknik Universitas Lampung.
6. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.

7. Segenap Dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat, wawasan, motivasi, dan pengalaman bagi Penulis.
8. Abyyu Hafizh Septrianto, teman hidup penulis yang selalu mencurahkan waktunya untuk mendukung dan membantu penulis pada seluruh pasang surut kehidupan, sehingga penulis mampu menjalani hidup di setiap harinya dengan rasa aman dan nyaman.
9. Alya Ananda Putri, Fannisa Nurhaffifi, dan Jena Feronika, sahabat seperjuangan selama masa perkuliahan yang selalu kebersamai, memberikan dukungan, dan membantu penulis dikala suka dan duka.
10. Shalsabila Erika dan Sabrina Early, sahabat penulis sejak dini yang selalu menemani setiap langkah, mendengarkan keluh kesah, memberikan dukungan, dan menjadi rumah kedua bagi penulis.
11. Kak Chairunisa, kakak yang selalu ada di setiap waktu untuk mendengar keluh kesah penulis, memberikan dukungan, dan menjadi pencari solusi dari semua pertanyaan.
12. Calista Putri Maharani, Fadhilah Wahyu Neeza, Elsy Mayonita, dan Badut, sahabat yang menemani perjalanan hidup, memberi rasa kasih sayang, dan memberi dukungan dimanapun mereka berada.
13. Sahabat Ksatria Himatro, Melda, Agita, Jakub, Ariska, Arya, Ziro, Alih, dan Cikal, yang selalu mendukung, menghibur, dan membuat penulis mendapatkan pengalaman berharga selama menjalani kepengurusan di Himatro.
14. Kepada keluarga Rectical angkatan 2022, yang telah memberikan warna baru dalam kehidupan penulis, khususnya selama masa perkuliahan.
15. Seluruh pihak yang telah membantu dan memberikan dukungannya yang tak dapat disebutkan satu persatu.

Bandar Lampung, 11 Mei 2026

Penulis,



Annisa Tri Hapsari

NPM. 2215061025

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	<b>i</b>
<b>DAFTAR GAMBAR</b> .....	<b>i</b>
<b>DAFTAR TABEL</b> .....	<b>i</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan Laporan.....	5
<b>II. TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1 <i>E-commerce</i> .....	7
2.2 <i>Back-end Development</i> .....	7
2.3 <i>Website</i> .....	8
2.4 Visual Studio Code.....	9
2.5 <i>Unified modelling language (UML)</i> .....	9
2.6 MySQL.....	11
2.7 <i>Framework Laravel</i> .....	12
2.8 REST API.....	12
2.9 PHP.....	13
2.10 Metode <i>Scrum</i> .....	14
2.10.1 <i>Scrum Team</i> .....	14
2.10.2 <i>Events Scrum</i> .....	16
2.10.3 <i>Artifacts Scrum</i> .....	17
2.11 <i>Black box Testing</i> .....	17
2.12 Postman .....	18

2.13 <i>Stress Test</i> .....	18
2.14 Apache JMeter.....	19
2.15 <i>Payment gateway</i> .....	19
2.16 MidTrans .....	20
2.17 Penelitian Terkait.....	21
<b>III. METODOLOGI PENELITIAN .....</b>	<b>26</b>
3.1 Waktu dan Tempat Penelitian .....	26
3.2 Perangkat Penelitian .....	26
3.3 Tahap Penelitian .....	28
3.3.1 Pengumpulan Data .....	30
3.3.2 <i>Product Backlog</i> .....	35
3.3.3 <i>Sprint Planning</i> .....	57
3.3.4 <i>Daily Scrum</i> .....	57
3.3.5 <i>Sprint Review</i> .....	58
3.3.6 <i>Sprint Retrospective</i> .....	59
3.3.7 Penulisan Laporan.....	59
<b>IV. HASIL DAN PEMBAHASAN .....</b>	<b>60</b>
4.1 Hasil.....	60
4.1.1 <i>Sprint 1</i> .....	60
4.1.2 <i>Sprint 2</i> .....	67
4.1.3 <i>Sprint 3</i> .....	88
4.1.4 <i>Sprint 4</i> .....	105
4.2 Pembahasan .....	125
4.2.1 Pengujian API Dengan Metode <i>Stress Test</i> .....	125
4.2.2 Capaian Penelitian.....	130
<b>V. SIMPULAN DAN SARAN.....</b>	<b>133</b>
5.1 Simpulan.....	133
5.2 Saran.....	134
<b>DAFTAR PUSTAKA .....</b>	<b>135</b>

## DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Metode <i>Scrum</i> .....	14
Gambar 3.1 Diagram Alir Penelitian. ....	29
Gambar 3.2 Diagram Batang Pernyataan Permasalahan.....	31
Gambar 3.3 Jawaban Responden Terkait Kesulitan Terbesar .....	31
Gambar 3.4 Jawaban Responden Terakit Fitur Paling Penting.....	32
Gambar 3.5 Diagram Batang kepentingan Elemen.....	32
Gambar 3.6 <i>Use case Diagram</i> U-Market .....	38
Gambar 3.7 <i>Activity Diagram</i> Registrasi .....	40
Gambar 3.8 <i>Activity Diagram</i> Mengelola Profil Admin.....	41
Gambar 3.9 <i>Activity Diagram</i> Mengelola Seluruh Produk .....	41
Gambar 3.10 <i>Activity Diagram</i> Mengelola Transaksi.....	42
Gambar 3.11 <i>Activity Diagram</i> Mengelola Pengguna .....	42
Gambar 3.12 <i>Activity Diagram</i> Mengakses Informasi Produk .....	43
Gambar 3.13 <i>Activity Diagram</i> Mengelola Profil Pengguna .....	43
Gambar 3.14 <i>Activity Diagram</i> Mengelola Keranjang .....	44
Gambar 3.15 <i>Activity Diagram</i> Melakukan Transaksi.....	45
Gambar 3.16 <i>Activity Diagram</i> Mengakses Status Pesanan .....	45
Gambar 3.17 <i>Activity Diagram</i> Mengakses Halaman Toko Saya.....	46
Gambar 3.18 <i>Activity Diagram Logout</i> .....	46
Gambar 3.19 <i>Class Diagram</i> U-Market.....	48
Gambar 3.20 <i>Mockup Landing page</i> .....	49
Gambar 3.21 <i>Mockup</i> Halaman <i>Login</i> .....	50
Gambar 3.22 <i>Mockup</i> Halaman Registrasi.....	50
Gambar 3.23 <i>Mockup</i> Halaman Beranda Admin .....	51
Gambar 3.24 <i>Mockup</i> Halaman Produk Admin.....	51
Gambar 3.25 <i>Mockup</i> Halaman Pengguna Admin.....	52

Gambar 3.26 <i>Mockup</i> Halaman Penjualan Admin.....	52
Gambar 3.27 <i>Mockup</i> Halaman Edit Profil Admin.....	53
Gambar 3.28 <i>Mockup</i> Halaman Beranda Pengguna .....	53
Gambar 3.29 <i>Mockup</i> Halaman Pesanan Saya.....	54
Gambar 3.30 <i>Mockup</i> Halaman Toko Saya .....	54
Gambar 3.31 <i>Mockup</i> Halaman Syarat & Ketentuan.....	55
Gambar 3.32 <i>Mockup</i> Halaman Edit Profil Pengguna.....	55
Gambar 3.33 <i>Mockup</i> Halaman Hubungi Kami.....	56
Gambar 3.34 <i>Mockup</i> Halaman Keranjang Saya .....	56
Gambar 4.1 Isi Dependensi Proyek <i>Website E-commerce U-Market</i> .....	61
Gambar 4.2 Struktur Folder <i>Back-end U-Market</i> .....	62
Gambar 4.3 Konfigurasi <i>Database</i> dengan <i>Environment Back-end</i> .....	64
Gambar 4.4 Pemodelan <i>Database</i> Tabel <i>Users</i> .....	64
Gambar 4.5 Pemodelan <i>Database</i> Tabel <i>Profiles</i> .....	65
Gambar 4.6 Pemodelan <i>Database</i> Tabel <i>Products</i> .....	65
Gambar 4.7 Hasil Rangkaian Acara <i>Sprint Retrospective Sprint 1</i> .....	67
Gambar 4.8 Pemodelan <i>Database</i> Tabel <i>Carts</i> .....	68
Gambar 4.9 Pemodelan <i>Database</i> Tabel <i>Cart Item</i> .....	69
Gambar 4.10 Pemodelan <i>Database</i> Tabel <i>Transactions</i> .....	69
Gambar 4.11 Pemodelan <i>Database</i> Tabel <i>Transaction Item</i> .....	70
Gambar 4.12 Penggunaan <i>Endpoint</i> <i>'/api/register'</i> dengan <i>Method</i> POST.....	71
Gambar 4.13 Penggunaan <i>Endpoint</i> <i>'/api/login'</i> dengan <i>Method</i> POST.....	71
Gambar 4.14 Penggunaan <i>Endpoint</i> <i>'/api/logout'</i> dengan <i>Method</i> POST.....	72
Gambar 4.15 Potongan Kode Fungsi <i>Hash::make ( )</i> .....	73
Gambar 4.16 Hasil Penerapan <i>Hash Password</i> .....	73
Gambar 4.17 Potongan Kode Pengecekan <i>Role</i> .....	74
Gambar 4.18 Pembatasan Akses Berdasarkan <i>Role</i> .....	74
Gambar 4.19 Penggunaan <i>Endpoint</i> <i>'/api/profile'</i> dengan <i>Method</i> GET .....	75
Gambar 4.20 Penggunaan <i>Endpoint</i> <i>'/api/profile'</i> dengan <i>Method</i> PUT.....	76
Gambar 4.21 Penggunaan <i>Endpoint</i> <i>'/api/Products'</i> dengan <i>Method</i> GET.....	77
Gambar 4.22 Penggunaan <i>Endpoint</i> <i>'/api/Products/{id}'</i> dengan <i>Method</i> GET ..	78

Gambar 4.23 Penggunaan <i>Endpoint</i> <code>'/api/Products/search'</code> dengan <i>Method</i> GET	79
Gambar 4.24 Penggunaan <i>Endpoint</i> <code>'/api/cart/add'</code> dengan <i>Method</i> POST	81
Gambar 4.25 Penggunaan <i>Endpoint</i> <code>'/api/cart'</code> dengan <i>Method</i> GET	82
Gambar 4.26 Penggunaan <i>Endpoint</i> <code>'/api/cart/remove/{item_id}'</code> dengan <i>Method</i> POST	83
Gambar 4.27 Penggunaan <i>Endpoint</i> <code>'/api/cart/count'</code> dengan <i>Method</i> GET	83
Gambar 4.28 Penggunaan <i>Endpoint</i> <code>'/api/cart/update'</code> dengan <i>Method</i> POST	84
Gambar 4.29 Hasil Rangkaian Acara <i>Sprint Retrospective Sprint 2</i>	87
Gambar 4.30 Pemodelan <i>Database</i> Tabel <i>Bank Account</i>	89
Gambar 4.31 Pemodelan <i>Database</i> Tabel <i>Withdrawals</i>	90
Gambar 4.32 Penggunaan <i>Endpoint</i> <code>'/api/checkout'</code> dengan <i>Method</i> POST	91
Gambar 4.33 Konfigurasi Integrasi <i>Payment gateway</i> Midtrans	92
Gambar 4.34 Penggunaan <i>Endpoint</i> <code>'/api/midtrans/notification'</code> dengan <i>Method</i> POST	93
Gambar 4.35 Penggunaan <i>Endpoint</i> <code>'/api/email/verify'</code> dengan <i>Method</i> POST	94
Gambar 4.36 Penggunaan <i>Endpoint</i> <code>'/api/admin/sales-data'</code> dengan <i>Method</i> GET	96
Gambar 4.37 Penggunaan <i>Endpoint</i> <code>'/api/admin/popular-Products'</code> dengan <i>Method</i> GET	96
Gambar 4.38 Penggunaan <i>Endpoint</i> <code>'/api/admin/cleanup-orphaned-items'</code> dengan <i>Method</i> GET	97
Gambar 4.39 Penggunaan <i>Endpoint</i> <code>'/api/admin/users'</code> dengan <i>Method</i> GET	98
Gambar 4.40 Penggunaan <i>Endpoint</i> <code>'/api/admin/users/{id}'</code> dengan <i>Method</i> POST	99
Gambar 4.41 Penggunaan <i>Endpoint</i> <code>'/api/admin/users/{id}'</code> dengan <i>Method</i> DELETE	99
Gambar 4.42 Penggunaan <i>Endpoint</i> <code>'/api/admin/transactions'</code> dengan <i>Method</i> GET	101
Gambar 4.43 Hasil Rangkaian Acara <i>Sprint Retrospective Sprint 3</i>	104
Gambar 4.44 Penggunaan <i>Endpoint</i> <code>'/midtrans/notification'</code> dengan <i>Method</i> POST	106

Gambar 4.45 Penggunaan <i>Endpoint</i> <code>'/api/store/{user_id}'</code> dengan <i>Method</i> GET .....	108
Gambar 4.46 Penggunaan <i>Endpoint</i> <code>'/api/store/{user_id}/Products'</code> dengan <i>Method</i> GET.....	108
Gambar 4.47 Penggunaan <i>Endpoint</i> <code>'/api/transactions'</code> dengan <i>Method</i> GET ..	110
Gambar 4.48 Implementasi Tombol Hubungi Penjual pada Halaman <i>Orders</i> ...	111
Gambar 4.49 Logika Pembentukan URL WhatsApp Berdasarkan Data Transaksi .....	112
Gambar 4.50 Penggunaan <i>Endpoint</i> <code>'/api/my-Products'</code> dengan <i>Method</i> GET..	114
Gambar 4.51 Penggunaan <i>Endpoint</i> <code>'/api/Products'</code> dengan <i>Method</i> POST.....	114
Gambar 4.52 Penggunaan <i>Endpoint</i> <code>'/api/Products/{id}'</code> dengan <i>Method</i> POST .....	115
Gambar 4.53 Penggunaan <i>Endpoint</i> <code>'/api/Products/{id}'</code> dengan <i>Method</i> DELETE .....	115
Gambar 4.54 Penggunaan <i>Endpoint</i> <code>'/api/user-banks'</code> dengan <i>Method</i> GET.....	116
Gambar 4.55 Penggunaan <i>Endpoint</i> <code>'/api/user-banks'</code> dengan <i>Method</i> POST...	116
Gambar 4.56 Penggunaan <i>Endpoint</i> <code>'/api/user-banks/{id}'</code> dengan <i>Method</i> DELETE.....	117
Gambar 4.57 Penggunaan <i>Endpoint</i> <code>'/api/seller-balance'</code> dengan <i>Method</i> GET	118
Gambar 4.58 Penggunaan <i>Endpoint</i> <code>'/api/seller-withdraw'</code> dengan <i>Method</i> POST .....	119
Gambar 4.59 Penggunaan <i>Endpoint</i> <code>'/api/user-withdrawals'</code> dengan <i>Method</i> GET .....	120
Gambar 4.60 Hasil Rangkaian Acara <i>Sprint Retrospective Sprint 4</i> .....	125
Gambar 4.61 Skenario <i>Stress Test</i> Pada Apache JMeter .....	127
Gambar 4.62 Membuat <i>Thread Group</i> .....	127

## DAFTAR TABEL

Tabel	Halaman
Tabel 3.1 Waktu Penelitian .....	26
Tabel 3.2 Perangkat Penelitian.....	27
Tabel 3.3.3 <i>Role Scrum</i> Pengembangan <i>Website U-Market</i> .....	28
Tabel 3.4 Kebutuhan Fungsional .....	33
Tabel 3.5 Kebutuhan Non Fungsional .....	34
Tabel 3.6 <i>Product Backlog</i> Pengguna (Fitur Utama).....	35
Tabel 3.7 <i>Product Backlog</i> Pengguna (Fitur Tambahan).....	36
Tabel 3.8 <i>Product Backlog</i> Admin (Fitur Utama) .....	36
Tabel 3.9 <i>Product Backlog</i> Admin (Fitur Tambahan) .....	36
Tabel 3.10 Deskripsi Aktor .....	37
Tabel 3.11 Definisi <i>Use case</i> .....	39
Tabel 4.1 <i>Sprint Backlog Sprint 1</i> .....	60
Tabel 4.2 Evaluasi <i>Task</i> Selama <i>Development</i> Pada Tahap <i>Sprint 1</i> .....	66
Tabel 4.3 <i>Sprint Backlog Sprint 2</i> .....	67
Tabel 4.4 <i>Black Box</i> API Autentikasi .....	85
Tabel 4.5 <i>Black Box Testing</i> API CRUD Profil.....	85
Tabel 4.6 <i>Black Box Testing</i> API Daftar Produk .....	86
Tabel 4.7 <i>Black Box Testing</i> API CRUD Keranjang .....	86
Tabel 4.8 Evaluasi <i>Task</i> Selama <i>Development Sprint 2</i> .....	87
Tabel 4.9 <i>Sprint Backlog Sprint 3</i> .....	88
Tabel 4.10 <i>Black Box Testing</i> API Checkout.....	102
Tabel 4.11 <i>Black Box Testing</i> API Verifikasi Email.....	102
Tabel 4.12 <i>Black Box Testing</i> API Aktivitas Admin .....	102
Tabel 4.13 Evaluasi <i>Task</i> Selama <i>Development Sprint 3</i> .....	104
Tabel 4.14 <i>Sprint Backlog Sprint 4</i> .....	105
Tabel 4.15 <i>Black Box Testing</i> API <i>Callback</i> Midtrans .....	121

Tabel 4.16 <i>Black Box Testing</i> API Riwayat Transaksi .....	122
Tabel 4.17 <i>Black Box Testing</i> API Halaman Penjual.....	122
Tabel 4.18 <i>Black Box Testing</i> API Manajemen Produk Penjual.....	122
Tabel 4.19 <i>Black Box Testing</i> API Manajemen Rekening Bank .....	123
Tabel 4.20 <i>Black Box Testing</i> API Manajemen Saldo dan Penarikan Dana.....	123
Tabel 4.21 <i>Sprint Review Sprint 4</i> .....	124
Tabel 4.22 Spesifikasi <i>Server</i> yang Digunakan dalam Melakukan <i>Stress Test</i> ..	126
Tabel 4.23 Hasil pengujian 150 <i>sample</i> dari Apache JMeter.....	128
Tabel 4.24 Hasil pengujian 200 <i>sample</i> dari Apache JMeter.....	128
Tabel 4.25 Hasil pengujian 250 <i>sample</i> dari Apache JMeter.....	128
Tabel 4.26 Hasil pengujian 300 <i>sample</i> dari Apache JMeter.....	128
Tabel 4.27 Rangkuman Hasil Pengujian <i>Stress Test</i> .....	129

## I. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia, termasuk dalam bidang perdagangan. Salah satu bentuk pemanfaatan teknologi informasi yang paling menonjol saat ini adalah *e-commerce*, yaitu sistem perdagangan elektronik yang memungkinkan penjual dan pembeli bertransaksi secara daring tanpa batasan ruang dan waktu. Model ini terbukti mampu meningkatkan efisiensi transaksi, memperluas jangkauan pasar, serta mempermudah pengelolaan data penjualan [1].

Di era digital saat ini, aktivitas jual beli tidak hanya dilakukan secara konvensional, tetapi telah beralih ke platform daring yang memungkinkan pelaku usaha maupun individu memasarkan produknya secara lebih luas. Di lingkungan Universitas Lampung, kegiatan jual beli sebelumnya banyak dilakukan melalui grup WhatsApp yang bersifat sementara dan tidak terstruktur. Meskipun praktis, cara ini memiliki berbagai kelemahan seperti informasi penjualan yang mudah tertimbun oleh pesan lain, tidak adanya sistem pencatatan data, serta kurangnya fitur keamanan transaksi [2]. Universitas Lampung memiliki potensi pasar yang besar dengan jumlah mahasiswa mencapai 44.839 orang, yang sebagian besar merupakan pengguna aktif internet dan perangkat digital. Kondisi ini menunjukkan peluang besar bagi terbentuknya ekosistem perdagangan daring yang tidak hanya dimanfaatkan oleh civitas akademika, tetapi juga dapat diakses oleh masyarakat umum.

Berdasarkan permasalahan dan potensi tersebut, dikembangkanlah sebuah *website e-commerce* bernama U-Market, yang dirancang untuk menjadi wadah jual beli daring yang dapat digunakan secara luas oleh masyarakat, termasuk mahasiswa,

pelaku usaha lokal, maupun pengguna umum. Platform ini dibangun menggunakan *framework* Laravel karena kemampuannya dalam menyediakan arsitektur *Model-View-Controller* (MVC) yang terstruktur, aman, serta mendukung pengembangan kode yang efisien [3]. Selain itu, proses pengembangan sistem dalam penelitian ini menggunakan metode *Scrum*, yaitu salah satu kerangka kerja Agile yang berfokus pada proses pengembangan iteratif dan inkremental melalui *Sprint* yang terstruktur. Pendekatan *Scrum* sangat efektif digunakan dalam pengembangan aplikasi web karena mampu meningkatkan fleksibilitas, mempercepat adaptasi terhadap perubahan kebutuhan, serta memastikan bahwa setiap *increment* yang dihasilkan dapat langsung diuji dan dievaluasi sebelum melanjutkan ke tahap berikutnya [4]. Dengan mekanisme ini, setiap fitur *backend* seperti autentikasi, manajemen produk, transaksi, maupun sistem pembayaran pada platform *e-commerce* U-Market dapat dikembangkan secara bertahap, diuji menggunakan Postman, dan disempurnakan melalui *Sprint Review* serta *retrospective* sehingga kualitas sistem dapat terus ditingkatkan selama proses pengembangan berlangsung.

Kebaruan dari penelitian ini terletak pada penerapan metode *Scrum* dalam pengembangan *back-end e-commerce* U-Market berbasis Laravel yang dirancang tidak hanya untuk lingkungan kampus, tetapi juga dapat dimanfaatkan secara luas oleh masyarakat umum, termasuk pelaku usaha lokal dan pengguna umum. Penerapan *Scrum* pada pengembangan *back end e-commerce* komunitas lokal masih relatif jarang dilakukan, khususnya dengan fokus pada pengelolaan sistem secara iteratif dan inkremental melalui pembagian pekerjaan ke dalam beberapa *Sprint*.

Penelitian ini menitikberatkan pada perancangan sisi *back end* untuk mengelola data produk, transaksi, dan pengguna secara aman, efisien, serta mudah dikembangkan, di mana setiap fitur dikembangkan secara bertahap, diuji, dan dievaluasi pada akhir *Sprint*. Dengan pendekatan tersebut, sistem dapat beradaptasi terhadap perubahan kebutuhan pengguna secara lebih fleksibel dan berkelanjutan. Oleh karena itu, penelitian ini diharapkan mampu memberikan kontribusi nyata dalam mendukung transformasi digital sektor perdagangan lokal sekaligus

memperkaya literatur terkait penerapan metode *Scrum* dan *framework* Laravel dalam pengembangan sistem *e-commerce* modern.

## 1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini, yaitu sebagai berikut:

1. Bagaimana merancang dan mengembangkan *back-end website e-commerce* U-Market dengan *framework* Laravel agar mampu memfasilitasi aktivitas jual beli secara daring bagi masyarakat umum secara terstruktur, efisien, dan mudah digunakan?
2. Bagaimana menerapkan metode *Scrum* dalam proses pengembangan sistem agar menghasilkan aplikasi dengan waktu yang relatif cepat dan sesuai dengan kebutuhan pengguna?
3. Bagaimana penerapan fitur-fitur *back end* seperti manajemen produk, autentikasi pengguna, dan pengelolaan transaksi dalam sistem U-Market?
4. Bagaimana efektivitas sistem U-Market dalam mengatasi masalah penumpukan informasi dan keterbatasan yang terjadi pada media jual beli berbasis grup WhatsApp?

## 1.3 Tujuan

Adapun tujuan dari penelitian ini diantaranya sebagai berikut:

1. Menghasilkan *back-end website e-commerce* U-Market yang dapat digunakan oleh masyarakat umum untuk melakukan aktivitas jual beli secara efisien dan aman.
2. Menerapkan *framework* Laravel untuk membangun sistem *back end* yang memiliki struktur terorganisasi, mudah dikembangkan, dan memiliki tingkat keamanan yang baik.
3. Mengimplementasikan metode *Scrum* untuk memastikan bahwa setiap *increment* yang dihasilkan dapat langsung diuji dan dievaluasi sebelum melanjutkan ke tahap berikutnya

4. Memberikan solusi terhadap permasalahan jual beli melalui grup WhatsApp dengan menghadirkan platform digital yang lebih terpusat dan terstruktur dengan jangkauan konsumen yang luas.

#### **1.4 Batasan Masalah**

Adapun batasan masalah dalam penelitian ini, yaitu sebagai berikut :

1. Penelitian ini hanya berfokus pada pengembangan sisi *back-end website e-commerce* U-Market menggunakan *framework* Laravel.
2. Metode pengembangan yang digunakan dalam penelitian ini adalah metode *Scrum*, yang menekankan pengembangan sistem secara iteratif melalui pembagian pekerjaan ke dalam beberapa *Sprint* untuk menyesuaikan sistem dengan kebutuhan pengguna.
3. Fitur keamanan yang dikembangkan meliputi autentikasi dan enkripsi dasar, tanpa pembahasan sertifikasi keamanan tingkat lanjut.

#### **1.5 Manfaat**

Adapun manfaat dari penelitian ini, yaitu sebagai berikut:

1. Menyediakan solusi alternatif bagi masyarakat umum khususnya warga Unila untuk melakukan aktivitas jual beli daring secara lebih terstruktur, efisien, dan aman dibandingkan dengan grup WhatsApp.
2. Mempermudah pengelolaan produk dan transaksi dengan sistem berbasis *database* yang terorganisasi.
3. Memberikan manfaat praktis berupa kemudahan bagi pengguna dalam bertransaksi digital melalui platform U-Market yang terintegrasi dan *user-friendly*.

## 1.6 Sistematika Penulisan Laporan

Laporan ini dibagi menjadi beberapa bab untuk memudahkan dalam penguraian, antara lain :

- BAB I** : PENDAHULUAN
- Bab ini memuat latar belakang yang menjelaskan alasan rancang bangun U-Market berbasis *website*, rumusan masalah yang ingin di selesaikan, tujuan penelitian, batasan masalah, manfaat yang diharapkan dari rancang bangun ini, serta sistematika penulisan laporan.
- BAB II** : TINJAUAN PUSTAKA
- Bab ini memuat teori dasar dalam rancang bangun U-Market. Pembahasan dalam bab ini meliputi *e-commerce*, *Back end Development*, *Website*, *Visual Studio Code*, *Unified modelling language (UML)*, *MySQL*, *Framework Laravel*, *REST API*, *PHP*, *Metode Scrum*, *Blackbox Testing*, *Postman*, *Stress Test*, *Apache Jmeter*, *Payment gateway*, *MidTrans* dan Penelitian Terkait.
- BAB III** : METODOLOGI PENELITIAN
- Bab ini membahas metodologi penelitian yang digunakan dalam rancang bangun U-Market, termasuk waktu dan tempat penelitian, perangkat keras dan perangkat lunak yang digunakan, serta tahapan penelitian yang dilakukan menggunakan metode *Scrum*.
- BAB IV** : HASIL DAN PEMBAHASAN
- Bab ini berisi hasil dari rancang bangun U-Market berbasis *website* menggunakan *framework Laravel*, serta pembahasan terkait implementasi sistem. Pembahasan mencakup desain sistem,

hasil pengujian fitur, dan analisis sistem dalam memenuhi kebutuhan pengguna, yaitu civitas di Universitas Lampung.

## BAB V

### : PENUTUP

Bab ini menyajikan kesimpulan dari penelitian yang telah dilakukan terkait rancang bangun U-Market, serta saran untuk pengembangan lebih lanjut yang dapat meningkatkan kinerja dan manfaat sistem bagi penggunanya.

## II. TINJAUAN PUSTAKA

### 2.1 *E-commerce*

*E-commerce* adalah rangkaian aktivitas bisnis yang memanfaatkan jaringan elektronik untuk melakukan transaksi jual-beli barang dan/atau jasa beserta dukungan proses pendukungnya, seperti pencatatan produk, katalog digital, manajemen pesanan, pembayaran elektronik, logistik, dan layanan purna-jual. *E-commerce* tidak hanya dipahami sebagai proses transaksi jual beli secara online, tetapi juga mencakup pertukaran informasi, pengelolaan hubungan pelanggan, serta integrasi proses bisnis secara digital. Perkembangan teknologi internet telah mengubah pola perdagangan tradisional menjadi sistem berbasis digital yang lebih efisien dan fleksibel [5]. Dalam arsitektur sistem, *e-commerce* tidak hanya soal tampilan toko online, melainkan sebuah ekosistem yang menggabungkan fungsi *front end* (interaksi pengguna) dan *back end* (pemrosesan transaksi, manajemen data, integrasi sistem pembayaran/kurir, keamanan, serta analitik) [6].

### 2.2 *Back-end Development*

*Backend development* merupakan proses pengembangan sisi *server* pada sebuah aplikasi berbasis web yang bertanggung jawab terhadap pengolahan data, logika bisnis, serta komunikasi antara aplikasi dengan basis data. Menurut Sarmanela, Samiya, dan Yuliana (2023), *backend development* mencakup pengelolaan *database*, pemrograman *server-side*, serta integrasi layanan melalui *Application Programming Interface* (API) yang berfungsi untuk mendukung kebutuhan sistem yang tidak terlihat secara langsung oleh pengguna. *Backend* menjadi fondasi utama

yang menentukan bagaimana data diproses, disimpan, dan disajikan kembali kepada pengguna melalui antarmuka aplikasi [7].

Dalam pengembangan aplikasi web modern, *backend development* berperan penting dalam menjaga stabilitas, keamanan, dan performa sistem. Proses *backend development* melibatkan penerapan mekanisme autentikasi dan otorisasi pengguna, pengelolaan transaksi data, serta penerapan validasi untuk memastikan integritas data yang disimpan. Selain itu, *backend development* juga harus mampu menangani permintaan dari pengguna secara efisien, terutama pada sistem yang memiliki aktivitas transaksi tinggi seperti *e-commerce*.

Pada sistem *e-commerce*, *backend development* memiliki peran strategis dalam mengelola data inti, seperti data produk, pengguna, dan transaksi. *Backend* harus dirancang agar mampu menangani proses transaksi secara aman, konsisten, dan terintegrasi dengan sistem *database*. Dalam penelitian ini, *backend development* diterapkan pada pengembangan sistem *e-commerce* U-Market berbasis *website* dengan memanfaatkan *framework* Laravel. Penggunaan Laravel dalam *backend development* membantu menciptakan struktur sistem yang terorganisir, aman, serta mudah dikembangkan lebih lanjut, sehingga mendukung kebutuhan sistem *e-commerce* yang dinamis dan kompleks.

### 2.3 Website

*Website* atau situs merupakan salah satu komponen paling fundamental dalam era digital saat ini, berfungsi sebagai media penyampaian informasi, sarana komunikasi, dan platform layanan interaktif bagi masyarakat. Sebuah *website* adalah gabungan dari beberapa halaman yang menampilkan informasi dalam berbagai bentuk seperti teks, gambar, audio, video, animasi, atau kombinasi dari semuanya yang saling terhubung dalam satu struktur dokumen yang dapat diakses melalui *World Wide Web* melalui alamat domain atau subdomain tertentu. *Website* memungkinkan konten bersifat dinamis dan interaktif, sekaligus membuka akses informasi secara global selama terhubung dengan internet [8].

Halaman-halaman pada *website* yang saling terhubung ini menjadikan media data yang menarik dan fungsional bagi pengguna. Struktur *website* yang terdiri atas teks, gambar, suara, dan aktivitas konten lainnya membutuhkan pengelolaan melalui protokol HTTP serta representasi halaman dalam HTML yang dapat ditangani oleh *server* web. Hal ini menunjukkan bahwa *website* bukan sekadar dokumen statis, melainkan merupakan himpunan *resource* yang dapat dikembangkan untuk berbagai tujuan, termasuk penyajian informasi, interaksi pengguna, dan penyediaan layanan informasi bisnis atau edukasi.

## 2.4 Visual Studio Code

Visual Studio Code merupakan sebuah editor kode sumber yang dikembangkan oleh Microsoft dan telah menjadi salah satu perangkat lunak pengembangan yang banyak digunakan oleh komunitas programmer modern. Dalam penelitian yang dilakukan oleh Hidayah dan Rofiqoh (2024), Visual Studio Code dijelaskan sebagai editor kode yang bersifat *open source*, ringan, dan mendukung berbagai fitur pengembangan seperti penyorotan sintaksis, penyelesaian kode otomatis, refaktor kode, *debugging*, serta integrasi dengan sistem kontrol versi seperti Git. Selain itu, antarmuka yang intuitif dan kemampuan penyesuaian melalui *extensions* membuat Visual Studio Code menjadi alat yang efektif dan efisien digunakan oleh pengembang perangkat lunak, terutama dalam pengembangan aplikasi berbasis web maupun *backend* seperti yang diterapkan dalam penelitian ini [9].

## 2.5 Unified modelling language (UML)

*Unified modelling language* (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain [10]. UML

menyediakan diagram-diagram seperti *use case*, aktivitas, kelas, *sequence*, dan lainnya yang memfasilitasi komunikasi antar tim dan mereduksi risiko kesalahpahaman dalam *requirement* dan desain sistem. UML digunakan untuk merancang model konseptual sistem, termasuk *use case diagram* untuk memetakan interaksi pengguna dan admin, diagram aktivitas untuk alur transaksi pembelian, serta diagram kelas untuk menggambarkan struktur basis data. Penggunaan UML ini membantu memperjelas kebutuhan sistem *e-commerce* sebelum proses pengkodean dilakukan.

a. *Use case diagram*

*Use case diagram* merupakan bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memodelkan kebutuhan fungsional sistem berdasarkan interaksi antara aktor dan sistem. *Use case diagram* membantu mengidentifikasi peran pengguna serta layanan yang disediakan sistem pada tingkat abstraksi yang tinggi sehingga mudah dipahami oleh pengguna maupun pengembang [11]. Dalam perancangan sistem *e-commerce* berbasis kemitraan, *use case diagram* digunakan untuk memetakan aktor seperti admin, mitra kerja, pengunjung, dan member beserta hak akses dan fungsinya masing-masing dalam sistem [12]. Dengan pendekatan tersebut, kebutuhan sistem dapat terdokumentasi secara sistematis sejak tahap analisis sehingga meminimalkan miskomunikasi selama proses pengembangan.

b. *Activity diagram*

*Activity diagram* digunakan untuk menggambarkan alur aktivitas atau proses bisnis yang berlangsung di dalam sistem secara lebih rinci. *Activity diagram* menunjukkan urutan kegiatan, percabangan keputusan, hingga kondisi awal dan akhir dari suatu proses, sehingga memudahkan analisis terhadap logika kerja sistem sebelum diimplementasikan [11]. Dalam implementasi sistem *e-commerce*, *activity diagram* dimanfaatkan untuk memodelkan proses seperti pengelolaan produk oleh admin, alur pemesanan oleh member, hingga proses konfirmasi pembayaran, sehingga setiap tahapan operasional dapat dirancang secara terstruktur [12]. Representasi

visual ini membantu memastikan bahwa alur sistem telah sesuai dengan kebutuhan bisnis yang direncanakan.

### c. *Class diagram*

*Class diagram* merepresentasikan struktur statis sistem dengan menampilkan kelas, atribut, metode, serta hubungan antar kelas dalam pendekatan berorientasi objek. UML menyediakan *class diagram* sebagai sarana untuk mendefinisikan arsitektur sistem secara konseptual sebelum tahap implementasi dilakukan [11]. Pada sistem *e-commerce* berbasis kemitraan, *class diagram* memuat entitas seperti Admin, Member, Produk, dan Kategori yang saling berelasi untuk mendukung proses transaksi dan pengelolaan data [12]. Dengan adanya pemodelan ini, struktur sistem menjadi lebih terorganisasi, memudahkan pengembangan, serta mendukung pemeliharaan dan pengembangan lanjutan di masa mendatang.

## 2.6 MySQL

*Database* merupakan komponen krusial dalam arsitektur *back-end* aplikasi *e-commerce* karena berfungsi sebagai pusat penyimpanan dan pengelolaan seluruh data yang digunakan sistem, seperti informasi produk, data pengguna, keranjang belanja, transaksi, serta histori pesanan. Menurut Connolly dan Begg [13], *database* adalah kumpulan data yang saling berhubungan yang disimpan secara sistematis agar dapat diakses dan dikelola secara efisien oleh perangkat lunak. Penggunaan sistem manajemen basis data (DBMS) memungkinkan pengembang untuk menerapkan operasi CRUD (*Create, Read, Update, Delete*) secara konsisten, serta menjaga integritas dan keamanan data dalam aplikasi.

MySQL merupakan salah satu DBMS yang paling umum digunakan dalam pengembangan aplikasi *web modern* karena bersifat *open-source*, mendukung bahasa SQL (*Structured Query Language*), dan memiliki performa tinggi pada transaksi berorientasi data besar. Menurut penelitian oleh Hidayat et al. [14], MySQL memiliki kemampuan optimasi *query* serta dukungan replikasi data yang

baik sehingga mampu menjaga kecepatan dan konsistensi sistem *e-commerce* berskala menengah hingga besar.

## 2.7 Framework Laravel

Laravel adalah *framework* PHP modern yang mengusung konsep MVC (*Model-View-Controller*) dan memudahkan pengembangan *web* dengan fitur seperti *routing*, ORM, *middleware*, dan *migration* [15]. MVC sendiri memiliki tiga bagian. Pertama, *model* berfungsi sebagai bagian yang menghubungkan aplikasi dengan *database*. Kedua, *view* berfungsi sebagai bagian dari desain *view* di mana *Controller* dikelola oleh *view*. Ketiga, *controller* berfungsi sebagai *controller* atau pengontrol *model* dan *view* sebelum selanjutnya menentukan aplikasi apa yang akan di proses.

Selain menerapkan arsitektur MVC, Laravel juga dilengkapi dengan berbagai fitur canggih yang mempercepat proses pengembangan aplikasi *web*, seperti *Blade Template Engine* untuk pengelolaan tampilan dinamis, *Eloquent ORM* untuk interaksi basis data yang lebih efisien, serta *Artisan CLI* yang memudahkan otomatisasi tugas-tugas rutin pengembang. *Framework* ini juga memiliki sistem keamanan yang baik melalui perlindungan *Cross-Site Request Forgery (CSRF)*, *Cross-Site Scripting (XSS)*, dan *SQL Injection*. Dengan dokumentasi yang lengkap serta dukungan komunitas yang luas, Laravel menjadi salah satu *framework* PHP paling populer dan andal dalam membangun aplikasi *web* berskala kecil hingga besar [1].

## 2.8 REST API

REST (*Representational State Transfer*) API adalah arsitektur komunikasi yang memungkinkan *frontend* atau aplikasi lain berinteraksi dengan *back end* menggunakan protokol HTTP dan format data JSON [16]. REST API digunakan sebagai penghubung antara sistem *back-end* Laravel dengan antarmuka *website*.

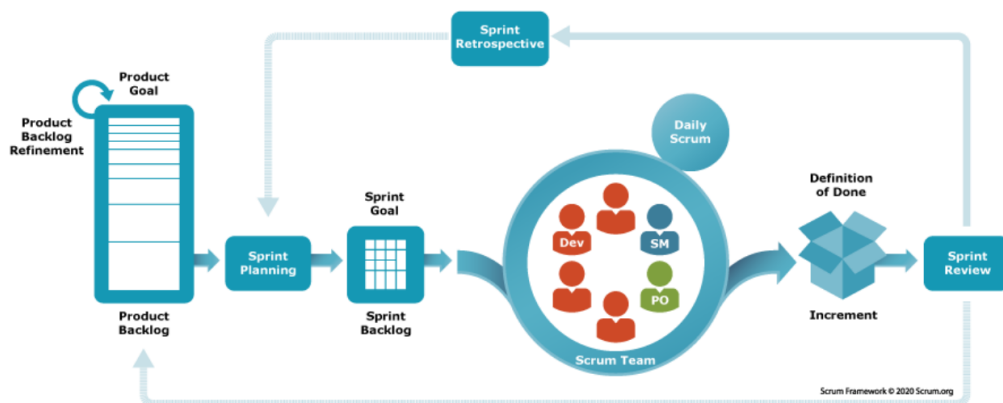
Setiap data produk, transaksi, dan pengguna dikomunikasikan melalui *endpoint* API yang terstruktur. Penggunaan REST API mendukung modularitas dan skalabilitas sistem, sehingga memudahkan integrasi dalam pengembangan. Selain itu, REST API memiliki prinsip kerja yang sederhana namun efisien karena bersifat stateless, artinya setiap permintaan (*request*) dari klien harus memuat seluruh informasi yang dibutuhkan tanpa bergantung pada permintaan sebelumnya. Hal ini membuat REST API mudah diimplementasikan dan diintegrasikan dengan berbagai platform seperti web, *mobile*, maupun aplikasi pihak ketiga. Dalam konteks pengembangan sistem berbasis Laravel, REST API sering digunakan untuk memisahkan antara logika *back end* dan tampilan *front end*, sehingga struktur aplikasi menjadi lebih modular dan mudah dikembangkan di masa mendatang.

## 2.9 PHP

PHP adalah bahasa pemrograman yang dirancang untuk menghasilkan halaman web secara interaktif di komputer yang menyajikannya, yang dikenal sebagai web *server*. Berbeda dengan HTML, di mana peramban web (*browser*) menggunakan *tag* dan *markup* untuk menampilkan halaman, kode PHP dijalankan di antara permintaan halaman dan *server* web, sehingga dapat menambahkan atau mengubah *output* HTML dasar [17]. Selain berfungsi sebagai bahasa pemrograman sisi *server* (*server-side scripting*), PHP juga memiliki kemampuan untuk berinteraksi dengan basis data, mengelola sesi pengguna, dan memproses formulir secara dinamis. PHP sering digunakan dalam pengembangan aplikasi web modern karena sifatnya yang *open source*, mudah dipelajari, serta memiliki kompatibilitas tinggi dengan berbagai sistem manajemen basis data seperti MySQL dan PostgreSQL. Dukungan komunitas yang luas serta integrasinya dengan berbagai *framework* menjadikan PHP tetap relevan sebagai fondasi utama pengembangan sistem berbasis web yang dinamis dan efisien.

## 2.10 Metode Scrum

Menurut *Scrum Guide*, *Scrum* adalah kerangka kerja yang membantu tim untuk mengembangkan dan mengelola produk dalam konteks kompleks melalui iterasi yang terstruktur. *Scrum* tidak menggantikan proses manajemen proyek tradisional, melainkan menyediakan fokus, transparansi, dan adaptasi sehingga pekerjaan yang dilakukan dapat terus dievaluasi dan disempurnakan dalam siklus kerja yang berkala. Kerangka kerja ini umumnya digunakan dalam konteks Agile, yang merupakan pendekatan pengembangan perangkat lunak yang berorientasi pada kolaborasi, nilai bisnis, dan responsif terhadap perubahan kebutuhan pengguna [18].



Gambar 2.1 Metode *Scrum* [18]

### 2.10.1 *Scrum Team*

*Scrum Team* merupakan tim yang dibentuk untuk mengembangkan produk secara kolaboratif dengan pendekatan adaptif. *Scrum Team* bekerja sebagai satu kesatuan yang saling melengkapi dalam mencapai tujuan pengembangan produk. Dalam kerangka kerja *Scrum*, *Scrum Team* terdiri atas tiga peran utama, yaitu *Product Owner*, *Scrum Master*, dan *Developers*. Setiap peran memiliki tanggung jawab yang berbeda namun saling terintegrasi untuk memastikan proses pengembangan berjalan efektif dan menghasilkan produk yang bernilai.

a. *Developers*

*Developers* merupakan anggota *Scrum Team* yang bertanggung jawab terhadap implementasi teknis dan realisasi produk. *Developers* dapat terdiri dari pengembang *front-end* maupun *back-end* yang bekerja sama dalam mengembangkan fitur sesuai dengan tujuan *Sprint*. Tanggung jawab *Developers* meliputi keterlibatan dalam proses *Sprint Planning* dan penyusunan *Sprint Backlog*, memastikan kualitas hasil pengembangan dengan mematuhi *definition of done* yang telah ditetapkan, serta melakukan penyesuaian rencana kerja secara harian agar tujuan *Sprint* dapat tercapai. Selain itu, *Developers* memiliki tanggung jawab bersama sebagai profesional untuk bekerja secara kolaboratif dan menjaga mutu produk yang dikembangkan.

b. *Product Owner*

*Product Owner* berperan sebagai pihak yang bertanggung jawab terhadap nilai produk yang dikembangkan oleh *Scrum Team*. *Product Owner* bertugas mengelola dan memprioritaskan *Product Backlog* agar pengembangan produk tetap selaras dengan kebutuhan pengguna dan tujuan proyek. Peran ini juga melibatkan komunikasi dengan pemangku kepentingan untuk memastikan bahwa kebutuhan dan harapan terhadap produk dapat diterjemahkan dengan baik ke dalam *item Sprint* yang dapat dikembangkan oleh tim.

c. *Scrum Master*

*Scrum Master* bertanggung jawab untuk memastikan bahwa penerapan *Scrum* berjalan sesuai dengan prinsip dan aturan yang telah ditetapkan. *Scrum Master* berperan sebagai fasilitator yang membantu *Scrum Team* memahami dan menerapkan *Scrum* secara efektif, serta menghilangkan hambatan yang dapat mengganggu proses pengembangan. Selain itu, *Scrum Master* memastikan setiap kegiatan *Scrum*, seperti *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*, dapat terlaksana dengan baik.

### 2.10.2 *Events Scrum*

*Scrum* membagi proses pengembangan ke dalam serangkaian kegiatan yang dikenal sebagai *events* atau tahapan, yaitu:

a. *Sprint Planning*

Tahapan ini merupakan pertemuan awal sebelum *Sprint* dimulai, di mana tim menentukan pekerjaan apa saja dari *Product Backlog* yang akan diselesaikan dalam *Sprint* tersebut. Perencanaan ini menghasilkan *Sprint Backlog* yang berisi tugas konkret untuk periode *Sprint* berikutnya.

b. *Sprint*

*Sprint* adalah periode kerja yang biasanya berdurasi 1–4 minggu, di mana tim bekerja untuk menghasilkan bagian produk yang dapat diuji (*increment*). Selama *Sprint*, tim memfokuskan tugasnya pada pekerjaan yang telah direncanakan tanpa perubahan besar dari luar.

c. *Daily Scrum*

*Daily Scrum* adalah rapat harian singkat (maksimal 15 menit) yang dilakukan setiap hari kerja selama *Sprint*. Tujuan rapat ini adalah untuk menyelaraskan rencana kerja harian, mengevaluasi progres, serta mengidentifikasi hambatan.

d. *Sprint Review*

Pada akhir *Sprint*, tim melakukan *Sprint Review* untuk mempresentasikan hasil yang telah dicapai kepada pemangku kepentingan (*stakeholders*). Hasil diskusi ini dapat memunculkan umpan balik yang kemudian diperhitungkan dalam *Sprint* berikutnya.

e. *Sprint Retrospective*

Tahapan ini dilakukan setelah *Sprint Review* dan sebelum *Sprint Planning* berikutnya. Tujuan *retrospective* adalah melakukan evaluasi proses kerja selama *Sprint* untuk meningkatkan efektivitas tim pada *Sprint* mendatang.

*Scrum* menawarkan beberapa kelebihan penting, seperti kemampuan untuk merespons perubahan kebutuhan dengan cepat, meningkatkan kolaborasi tim, serta menyediakan mekanisme evaluasi berkala melalui *Sprint*. Pendekatan iteratif ini membantu tim untuk mengurangi risiko kegagalan dengan memperkecil ruang lingkup pengerjaan sekaligus mempercepat *feedback* dari pengguna atau pemangku kepentingan. Faktor-faktor ini menjadikan *Scrum* sangat populer, terutama dalam pengembangan sistem yang kompleks seperti aplikasi web dan *backend* berbasis API yang memerlukan adaptasi terhadap permintaan pasar yang berubah-ubah.

### 2.10.3 *Artifacts Scrum*

*Artifacts* atau luaran yang dihasilkan pada setiap proses *Scrum* adalah sebagai berikut:

a. *Product Backlog*

Merupakan daftar prioritas dari seluruh kebutuhan fitur, perbaikan, dan perubahan yang harus dikerjakan dalam pengembangan produk. *Product Backlog* disusun oleh *Product Owner* dan menjadi dasar pengaturan pekerjaan dalam *Sprint*.

b. *Sprint Backlog*

Merupakan daftar *Product Backlog* yang dipilih untuk *Sprint* ditambah rencana untuk memproduksi produk tersebut dan mencapai tujuan *Sprint*.

c. *Increment*

Merupakan jumlah *item Product Backlog* yang diselesaikan dalam *Sprint* dan nilai total *increment* di semua *Sprint* sebelumnya.

### 2.11 *Black box Testing*

*Black-box testing* merupakan salah satu metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa memperhatikan struktur internal atau logika

program yang digunakan untuk menghasilkan keluaran tersebut. Pengujian jenis ini dilakukan dengan tujuan memastikan bahwa perangkat lunak dapat berfungsi sesuai dengan kebutuhan pengguna (*user requirements*) yang telah ditentukan pada tahap analisis. Penguji hanya mengetahui masukan (*input*) dan keluaran (*output*) yang diharapkan, tanpa perlu memahami bagaimana proses internal atau kode sumber bekerja untuk mencapai hasil tersebut. Pendekatan *black-box testing* cocok diterapkan untuk mengidentifikasi kesalahan pada fungsi, antarmuka, serta kinerja sistem yang berhubungan langsung dengan pengguna. Selain itu, metode ini efektif untuk mendeteksi kesalahan pada logika program, kesalahan implementasi kebutuhan, serta masalah kompatibilitas antar komponen [19].

## 2.12 Postman

Postman adalah sebuah alat pengujian *Application Programming Interface* (API) yang memungkinkan pengembang dan penguji perangkat lunak untuk mengirim permintaan HTTP/HTTPS, menerima tanggapan dari *server*, serta menjalankan skrip pengujian, semuanya dalam satu antarmuka yang terintegrasi. Artikel oleh Kore et al. (2022) menekankan bahwa Postman telah berkembang dari *plugin browser* sederhana menjadi solusi otomatisasi dan dokumentasi API yang digunakan oleh jutaan pengembang dan ratusan ribu organisasi [20]. Postman mendukung berbagai jenis pengujian seperti pengujian fungsional, integrasi, performa, dan keamanan API melalui fitur koleksi (*collections*), variabel lingkungan (*environment variables*), skrip *pre-request* dan *post-response*, serta integrasi *pipeline* CI/CD. Contoh panduan resmi Postman menyebutkan bahwa pengujian API mencakup *integration testing*, *end-to-end testing*, *regression testing*, dan *performance testing*.

## 2.13 Stress Test

*Stress Testing* adalah cara untuk melihat bagaimana situasi rawan kegagalan mempengaruhi tingkat stres sistem. Pengujian akan dilakukan dengan variabel

jumlah pengguna dalam satu waktu dan akan menguji respon sistem ketika diakses oleh pengguna dalam jumlah besar dan dalam jangka waktu tertentu [21]. Tujuan dari *Stress Testing* adalah untuk memperkirakan beban maksimum yang dapat ditangani oleh *server* web.

#### **2.14 Apache JMeter**

Apache JMeter adalah proyek sumber terbuka yang digunakan untuk alat pengujian kinerja dan beban. Apache JMeter akan membuat beberapa simulasi pengguna yang akan mengakses *server* dengan jumlah pengguna yang berbeda dan interval waktu permintaan tergantung pada konfigurasi yang dilakukan pada Apache JMeter [21].

#### **2.15 Payment gateway**

*Payment gateway* merupakan layanan perantara dalam sistem transaksi elektronik yang berfungsi menghubungkan pelanggan, *merchant*, dan lembaga perbankan dalam proses pembayaran daring secara aman. Dalam praktik *e-commerce*, *payment gateway* berperan sebagai pihak ketiga tepercaya (*trusted third party*) yang menerima, mengenkripsi, serta meneruskan data pembayaran pelanggan untuk proses otorisasi tanpa harus mengekspos informasi sensitif kepada *merchant* secara langsung. Konsep ini antara lain dijelaskan oleh Kyaw Zay Oo dalam artikelnya yang berjudul *Design and Implementation of Electronic Payment gateway for Secure Online Payment System* [22], yang menekankan pentingnya mekanisme enkripsi dan validasi dalam menjaga kerahasiaan serta integritas data transaksi. Dengan memanfaatkan protokol keamanan seperti SSL/TLS dan algoritma kriptografi kunci publik, *payment gateway* tidak hanya memastikan proses otorisasi berjalan secara *real time*, tetapi juga meminimalkan risiko penyalahgunaan data, penipuan, serta sengketa transaksi.

*Payment gateway* dalam implementasinya tidak hanya berfungsi sebagai penghubung teknis antara sistem *merchant* dan lembaga keuangan, tetapi juga

menjadi bagian integral dari arsitektur sistem informasi yang mendukung efisiensi dan keandalan transaksi daring. Integrasi *payment gateway*, seperti Midtrans, memungkinkan proses verifikasi pembayaran dilakukan secara otomatis melalui mekanisme notifikasi (*callback*) dan API, sehingga meminimalkan kesalahan pencatatan serta mempercepat konfirmasi transaksi. Selain itu, sistem ini menyediakan berbagai metode pembayaran dalam satu platform terintegrasi, yang pada akhirnya meningkatkan fleksibilitas serta kenyamanan pengguna dalam bertransaksi. Penerapan tersebut menunjukkan bahwa digitalisasi sistem pembayaran melalui *payment gateway* mampu meningkatkan efektivitas pengelolaan transaksi dan mengurangi ketergantungan pada proses manual [23].

## 2.16 MidTrans

Midtrans merupakan salah satu penyedia layanan *payment gateway* terkemuka di Indonesia yang berfungsi sebagai perantara antara penjual dan berbagai lembaga keuangan untuk memfasilitasi transaksi daring secara aman dan efisien. Melalui Midtrans, pengguna dapat melakukan pembayaran menggunakan beragam metode seperti kartu kredit, *virtual account*, dompet digital, maupun QRIS, tanpa perlu melakukan integrasi langsung dengan setiap penyedia layanan pembayaran. Layanan ini membantu pelaku usaha maupun pengembang sistem *e-commerce* dalam mempercepat proses transaksi sekaligus meningkatkan kepercayaan pelanggan terhadap sistem pembayaran digital [24]. Dokumentasi resmi Midtrans menjelaskan bahwa sistem ini dibangun untuk mendukung berbagai model integrasi, seperti *snap* untuk integrasi cepat berbasis *hosted payment page*, *Core API* untuk integrasi kustom berbasis REST API, serta *Webhook Notification* untuk mengirimkan status transaksi secara *real-time* dari *server* Midtrans ke *server merchant*.

Dalam aspek keamanan, Midtrans telah memperoleh sertifikasi *Payment Card Industry Data Security Standard* (PCI-DSS) Level 1, yang merupakan standar keamanan tertinggi dalam pengelolaan data kartu pembayaran. Selain itu, Midtrans menerapkan teknologi *Advanced Encryption Standard* (AES-256) serta sistem

deteksi penipuan berbasis *machine learning* bernama Aegis guna menjaga keamanan transaksi dan mencegah terjadinya aktivitas mencurigakan. Keunggulan tersebut menjadikan Midtrans banyak digunakan pada berbagai platform *e-commerce* di Indonesia karena keandalannya dalam menangani proses otorisasi, verifikasi, dan notifikasi transaksi secara otomatis.

Pengembangan sistem *payment gateway* berbasis Midtrans memperlihatkan kontribusi signifikan terhadap modernisasi layanan berbasis digital, khususnya dalam konteks transformasi sistem pembayaran konvensional menuju sistem elektronik yang lebih adaptif. Dengan memanfaatkan enkripsi data dan protokol keamanan yang terstandarisasi, integrasi ini mampu menjaga keamanan data transaksi sekaligus memastikan proses otorisasi berjalan secara *real time*. Implementasi tersebut tidak hanya meningkatkan keamanan, tetapi juga memperkuat kepercayaan pengguna terhadap sistem pembayaran digital yang digunakan. Hasil pengembangan yang dilakukan dalam penelitian terkait menunjukkan bahwa penggunaan *payment gateway* mendukung peningkatan kualitas layanan dan efisiensi operasional pada sistem yang terdigitalisasi [25].

## 2.17 Penelitian Terkait

Berdasarkan berbagai penelitian sebelumnya, metode *Scrum* dan *framework* Laravel telah banyak digunakan dalam pengembangan sistem informasi dan aplikasi *e-commerce* berbasis web. Tinjauan terhadap beberapa penelitian berikut bertujuan untuk memperkuat dasar teori dan memberikan gambaran mengenai relevansi metode serta teknologi yang digunakan dalam penelitian ini.

Penelitian yang dilakukan oleh Fauzi dan Darmawan (2023) membahas pembangunan aplikasi *e-commerce* berbasis *website* menggunakan *framework* Laravel untuk mendukung aktivitas penjualan UMKM. Penelitian ini menekankan pemanfaatan Laravel dalam membangun sistem yang terstruktur, aman, dan efisien dengan menerapkan konsep MVC. Sistem yang dikembangkan mampu menangani proses promosi produk, pemesanan, hingga pembayaran secara daring. Hasil

penelitian menunjukkan bahwa penggunaan Laravel dapat mempercepat proses pengembangan *website e-commerce* serta mempermudah pengelolaan data transaksi dan produk secara terpusat. Penelitian ini menjadi rujukan penting dalam pengembangan sistem *e-commerce* berbasis web yang berfokus pada pengelolaan *back-end* yang terstruktur [24].

Penelitian selanjutnya dilakukan oleh Andipradana dan Hartomo (2021) yang merancang dan membangun aplikasi penjualan online berbasis web dengan menerapkan metode *Scrum* sebagai pendekatan pengembangan sistem. Penelitian ini menyoroti fleksibilitas *Scrum* dalam menangani perubahan kebutuhan sistem selama proses pengembangan. Aplikasi yang dihasilkan mampu mendukung pengelolaan produk, transaksi, serta pelaporan penjualan secara digital. Hasil penelitian menunjukkan bahwa metode *Scrum* efektif digunakan pada pengembangan sistem penjualan online karena memungkinkan iterasi cepat dan kolaborasi intensif antar tim pengembang, sehingga sistem yang dihasilkan lebih adaptif terhadap kebutuhan pengguna [26].

Penelitian lain yang dilakukan oleh Syahputra dkk. (2024) membahas rancang bangun sistem informasi penjualan mainan edukasi berbasis web menggunakan metode *Scrum*. Penelitian ini menekankan pembagian proyek ke dalam beberapa *Sprint* untuk menghasilkan fitur secara bertahap. Sistem yang dikembangkan mampu meningkatkan efisiensi pencatatan transaksi, pengelolaan stok, serta penyusunan laporan penjualan. Hasil pengujian menunjukkan bahwa sistem memiliki tingkat fungsionalitas yang baik dan mudah digunakan. Penelitian ini memperkuat bahwa metode *Scrum* sangat sesuai untuk pengembangan sistem informasi penjualan berbasis web yang membutuhkan fleksibilitas dan evaluasi berkelanjutan [27].

Rahmouni dkk. (2023) dalam penelitiannya membahas pendekatan pemodelan untuk menghasilkan kode aplikasi *e-commerce* berbasis Laravel dengan menerapkan *Model Driven Architecture* (MDA). Penelitian ini menjelaskan bagaimana pemodelan sistem menggunakan UML dapat ditransformasikan secara

sistematis menjadi kode program Laravel yang mengikuti arsitektur MVC. Hasil penelitian menunjukkan bahwa pendekatan pemodelan ini mampu meningkatkan konsistensi desain sistem dan mempercepat proses pengembangan *back-end* aplikasi *e-commerce*. Penelitian ini relevan sebagai landasan konseptual dalam perancangan arsitektur *back-end* yang terstruktur dan mudah dikembangkan [28].

Penelitian oleh Aditya dkk. (2022) membahas pengembangan sistem *e-commerce* dengan pemanfaatan *payment gateway* Midtrans menggunakan *framework* Laravel. Penelitian ini difokuskan pada otomatisasi proses pembayaran yang sebelumnya dilakukan secara manual. Hasil penelitian menunjukkan bahwa integrasi Midtrans mampu mempercepat proses verifikasi pembayaran dan meningkatkan keandalan sistem transaksi. Selain itu, pengujian sistem menggunakan metode *blackbox* menunjukkan bahwa seluruh fungsi sistem berjalan dengan valid. Penelitian ini menjadi referensi penting dalam pengembangan fitur *checkout* dan manajemen transaksi pada sistem *e-commerce* berbasis *back-end* [29].

Penelitian selanjutnya dilakukan oleh Ayurira dan Fajri (2024) yang mengimplementasikan metode *Scrum* dalam pengembangan *website e-commerce* pada Twins Petshop. Penelitian ini menyoroti keunggulan *Scrum* dalam meningkatkan kolaborasi tim dan responsivitas sistem terhadap perubahan kebutuhan. Sistem yang dikembangkan diuji menggunakan metode *blackbox* dan menunjukkan hasil fungsional yang baik. Penelitian ini menegaskan bahwa metode *Scrum* efektif diterapkan pada pengembangan *website e-commerce* untuk bisnis skala kecil dan menengah, khususnya dalam menghasilkan sistem yang adaptif dan berkualitas [30]

Penelitian oleh Sinaga dkk. (2021) membahas implementasi *framework* Laravel dengan konsep MVC dalam pengembangan aplikasi *e-commerce* berbasis web. Penelitian ini menekankan pemisahan antara *model*, *view*, dan *controller* untuk menghasilkan struktur kode yang rapi dan mudah dikelola. Sistem yang dikembangkan mendukung berbagai peran pengguna serta pengelolaan data transaksi dan produk. Hasil penelitian menunjukkan bahwa penggunaan Laravel

dengan arsitektur MVC dapat meningkatkan efisiensi pengembangan dan pemeliharaan sistem *e-commerce*, khususnya pada sisi *back-end* aplikasi [31].

Berdasarkan penelitian terkait yang telah diuraikan, dapat disimpulkan bahwa sebagian besar penelitian telah berhasil membangun sistem *e-commerce* berbasis *website* dengan memanfaatkan *framework* Laravel sebagai kerangka kerja pengembangan aplikasi. Penelitian-penelitian tersebut umumnya berfokus pada implementasi fitur dasar *e-commerce*, seperti pengelolaan produk, transaksi, dan laporan penjualan, serta menekankan keunggulan Laravel dalam mendukung arsitektur MVC yang terstruktur dan mudah dikembangkan. Selain itu, beberapa penelitian juga telah menerapkan metode *Scrum* sebagai pendekatan pengembangan sistem untuk menghasilkan aplikasi yang adaptif terhadap perubahan kebutuhan pengguna.

Namun demikian, masih terdapat beberapa celah penelitian (*research gap*) yang dapat diidentifikasi. Pertama, sebagian penelitian lebih menitikberatkan pada pengembangan sistem secara umum tanpa memberikan penjelasan yang mendalam mengenai perancangan dan implementasi sisi *back-end* sebagai fondasi utama sistem *e-commerce*. Kedua, integrasi layanan REST API sebagai penghubung antara *front-end* dan *back-end* belum dibahas secara komprehensif, khususnya terkait pengelolaan data, validasi, serta keamanan akses API. Ketiga, meskipun terdapat penelitian yang mengintegrasikan *payment gateway*, pembahasan mengenai alur pembayaran, pengelolaan status transaksi, serta mekanisme *callback* dari *payment gateway* pada sisi *back-end* masih relatif terbatas. Selain itu, penerapan metode *Scrum* pada penelitian terdahulu umumnya belum dikaitkan secara rinci dengan penyusunan *Product Backlog*, pembagian *Sprint*, dan evaluasi hasil pengembangan pada setiap *Sprint* secara sistematis.

Berdasarkan gap penelitian tersebut, penelitian ini dilakukan untuk melengkapi dan mengembangkan penelitian-penelitian sebelumnya dengan memfokuskan pada rancang bangun *back-end* sistem *e-commerce* berbasis *website*. Penelitian ini menitikberatkan pada pengembangan sisi *back-end* menggunakan *framework*

Laravel dengan pendekatan REST API, serta menerapkan metode *Scrum* secara terstruktur melalui penyusunan *Product Backlog* dan pelaksanaan *Sprint* secara bertahap. Selain itu, penelitian ini juga mengintegrasikan *payment gateway* Midtrans untuk mendukung proses checkout dan pengelolaan transaksi secara otomatis. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem *e-commerce* yang lebih terstruktur, aman, dan mudah dikembangkan, sekaligus memperkaya kajian ilmiah terkait pengembangan *back-end e-commerce* berbasis Laravel dan *Scrum*.

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan dalam kurun waktu 6 bulan, yaitu pada bulan November 2025 hingga April 2026 di Laboratorium Terpadu Teknik Elektro, Jurusan Teknik Elektro, Universitas Lampung.

Tabel 3.1 Waktu Penelitian

No.	Kegiatan Penelitian	Nov 2025	Des 2025	Jan 2026	Feb 2026	Mar 2026	Apr 2026
1.	Studi Literatur						
2.	Pengumpulan Data						
2.	<i>Product Backlog</i>						
3.	<i>Sprint 1</i>						
4.	<i>Sprint 2</i>						
5.	<i>Sprint 3</i>						
6.	<i>Sprint 4</i>						
7.	Penulisan Laporan						

#### 3.2 Perangkat Penelitian

Adapun alat yang digunakan pada penelitian dan pengembangan sistem yang dibuat adalah sebagai berikut:

Tabel 3.2 Perangkat Penelitian

No.	Perangkat	Spesifikasi	Kegunaan
1	Laptop	DELL Latitude E7450 Desktop-U8VO61R, 8GB RAM, 512GB Storage, Windows 10	Perangkat utama untuk pengembangan sistem
2	<i>Framework</i> Laravel	Versi 12	Digunakan untuk membangun aplikasi berbasis web.
3	<i>Database Management System</i>	MySQL 8.0	Digunakan untuk penyimpanan dan pengelolaan data.
4	<i>Web server</i>	Apache	<i>Server</i> untuk menjalankan aplikasi web.
5	<i>Version Control System</i>	Git & GitHub	Digunakan untuk kontrol versi dan kolaborasi pengembangan.
6	<i>Editor Code</i>	Visual Studio Code dengan ekstensi Laravel	Digunakan sebagai IDE utama dalam pengembangan <i>back end</i> .
7	<i>Browser</i>	Google Chrome, Microsoft Edge	Digunakan untuk pengujian Tampilan dan responsivitas sistem.
8	<i>Diagramming Tools</i>	Draw.io, DBdiagram.io	Digunakan untuk membuat diagram arsitektur sistem dan <i>database</i> .
9	<i>Project Management Tools</i>	Trello, Retros Work	Digunakan untuk mengelola <i>Product Backlog</i> , <i>Sprint Backlog</i> , dan memantau progres pengembangan menggunakan metode <i>Scrum</i> .
10	<i>UI/UX Design Tools</i>	Figma	Digunakan untuk perancangan antarmuka pengguna dan <i>wireframe</i> sistem
11	<i>API Testing Tools</i>	Postman	Digunakan untuk pengujian REST API <i>back-end</i> , validasi <i>request-response</i> , serta pengujian fungsional sistem.
12	<i>Stress Testing Tools</i>	Apache JMeter	Digunakan untuk pengujian beban sistem dalam simulasi banyaknya pengguna.

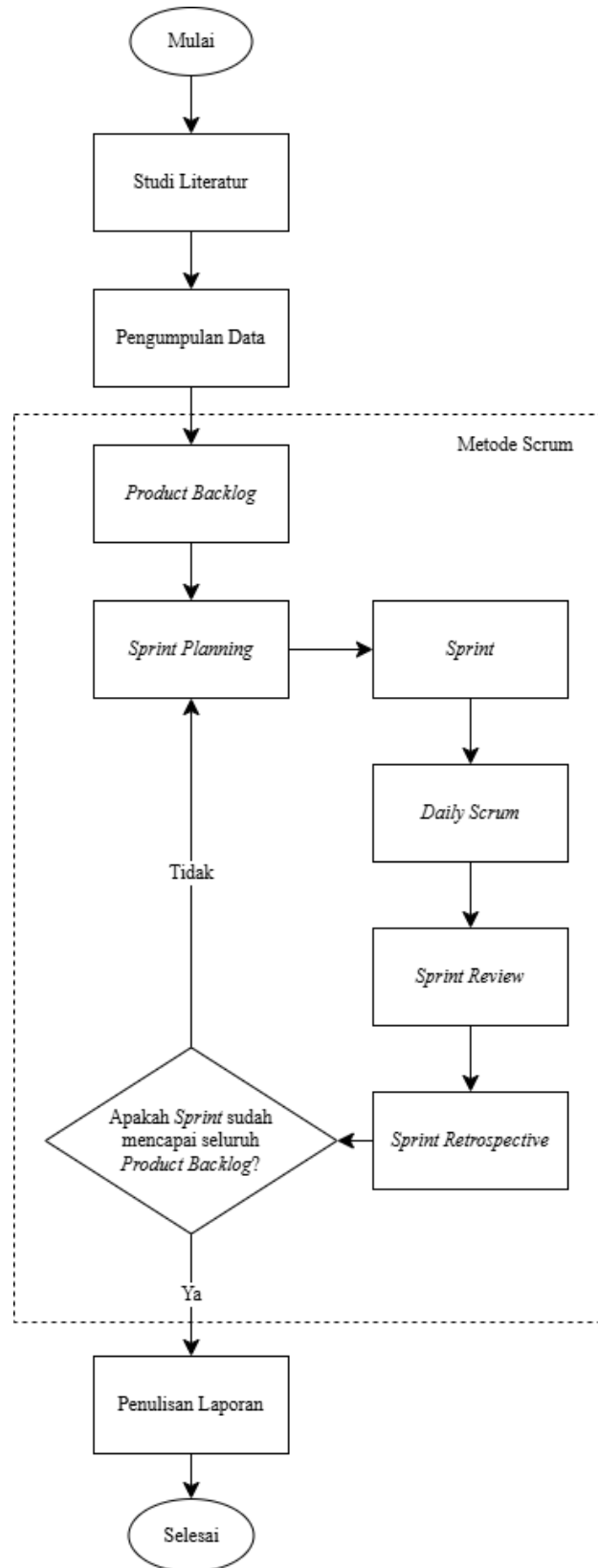
### 3.3 Tahap Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang disusun secara sistematis agar tujuan penelitian dapat tercapai dengan baik. Tahapan dimulai dengan studi literatur hingga mengaplikasikan metode *Scrum*, yang berfungsi sebagai visualisasi alur kerja dan membantu dalam membatasi ruang lingkup proses yang sedang berlangsung. Metode ini sesuai saat digunakan dalam pengembangan perangkat lunak secara tim karena dapat menjaga koordinasi yang baik antar masing-masing anggota tim yang memiliki tugas yang berbeda. Tim-tim yang terdapat pada penelitian ini adalah sebagai berikut

Tabel 3.3 *Role Scrum* Pengembangan *Website* U-Market

<i>Role</i>	Nama	Keterangan
<i>Product Owner</i>	Yessi Mulyani, S.T., M.T.	<i>Owner</i> Platform U Marketplace. Tugas dari <i>Product Owner</i> dalam penelitian ini adalah untuk memastikan bahwa <i>developer website</i> U-Market memenuhi kebutuhan pengguna platform U Marketplace.
<i>Scrum Master</i>	Nurrahma, M.T.	<i>Scrum Master</i> bertanggung jawab untuk memperkenalkan dan mendukung implementasi Metode <i>Scrum</i> agar dapat berjalan dengan baik.
<i>Developers</i>	Alya Ananda Putri	<i>Front-end Developer</i>
	Annisa Tri Hapsari	<i>Back-end Developer</i>

Selain pembagian peran dalam tim, penelitian ini juga menekankan pentingnya kolaborasi dan komunikasi yang berkelanjutan antar anggota tim selama proses pengembangan berlangsung. Setiap tahapan dalam metode *Scrum* dilaksanakan secara iteratif dan inkremental, sehingga memungkinkan adanya evaluasi dan perbaikan secara berkala pada setiap akhir *Sprint*. Dengan pendekatan ini, pengembangan sistem tidak hanya berfokus pada penyelesaian fitur, tetapi juga pada peningkatan kualitas perangkat lunak serta kesesuaian dengan kebutuhan pengguna. Adapun tahapan-tahapan yang dilakukan dalam pengembangan *website e-commerce* U-Market disajikan dalam diagram berikut:



Gambar 3.1 Diagram Alir Penelitian.

### 3.3.1 Pengumpulan Data

Pengumpulan data merupakan tahap awal yang bertujuan untuk mengidentifikasi kebutuhan sistem serta memperoleh pemahaman mendalam mengenai proses bisnis yang akan diotomatisasi. Pada tahap ini dilakukan pengumpulan data melalui beberapa teknik agar kebutuhan pengguna dan sistem dapat diidentifikasi secara akurat. Metode pengumpulan data yang digunakan pada penelitian ini meliputi observasi langsung dan wawancara terhadap pihak yang terkait dengan sistem *e-commerce* yang akan dibangun. Bentuk pengumpulan data yang dilakukan adalah sebagai berikut:

#### 1. Observasi

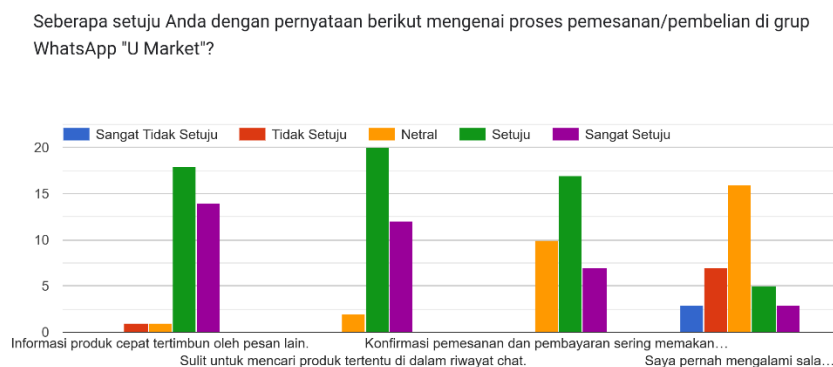
Tahap observasi bertujuan untuk mengidentifikasi proses bisnis *e-commerce* yang berjalan serta kebutuhan pengguna terhadap sistem. Pada tahap ini peneliti melakukan pengamatan langsung terhadap alur transaksi, pengelolaan produk, dan aktivitas penjual maupun pembeli pada sistem *e-commerce* serupa. Selain itu, dilakukan pula observasi terhadap kebutuhan pengelolaan data oleh admin untuk memahami bagaimana sistem *back-end* akan dirancang.

Berdasarkan hasil observasi terhadap sistem *e-commerce* serupa, ditemukan bahwa proses pengelolaan produk, transaksi, dan data pengguna memerlukan efisiensi yang tinggi. Sebagian besar sistem menyediakan pengelolaan *back-end* yang fleksibel bagi admin untuk memantau data secara *real time*. Selain itu, dibutuhkan REST API yang stabil agar sistem *frontend* dapat berkomunikasi dengan *server* tanpa gangguan. Dari hasil pengamatan ini, ditetapkan bahwa *back-end* U-Market harus mampu menangani data produk, transaksi, dan pengguna secara terpusat dengan tingkat keamanan yang tinggi.

#### 2. Wawancara

Tahap wawancara bertujuan untuk menggali informasi mendalam mengenai kebutuhan pengguna dan fitur yang diharapkan dari sistem. Pada tahap ini peneliti melakukan wawancara dengan beberapa calon pengguna, seperti admin, penjual,

dan pembeli, untuk mengetahui kebutuhan, harapan, serta permasalahan yang sering muncul dalam pengelolaan transaksi *online*. Berdasarkan hasil wawancara yang dilakukan *online* melalui media Google Formulir, terdapat 34 responden dengan rentan usia 19 – 44 Tahun (Kategori Dewasa) yang memberikan jawabannya, dimana 26 reponden sebagai pembeli U-Market di platform WhatsApp, 5 responden sebagai penjual, dan 3 responden sebagai penjual dan pembeli. Sebanyak 32,4% responden menyatakan bahwa mereka melakukan pembelian di U-Market melalui Whatsapp lebih dari 1 kali dalam 1 bulan dengan jenis produk makanan dan jasa. Namun, terdapat beberapa kendala dalam proses transaksi yang dilakukan melalui WhatsApp, kendala tersebut dapat dilihat dalam diagram berikut.



Gambar 3.2 Diagram Batang Pernyataan Permasalahan.

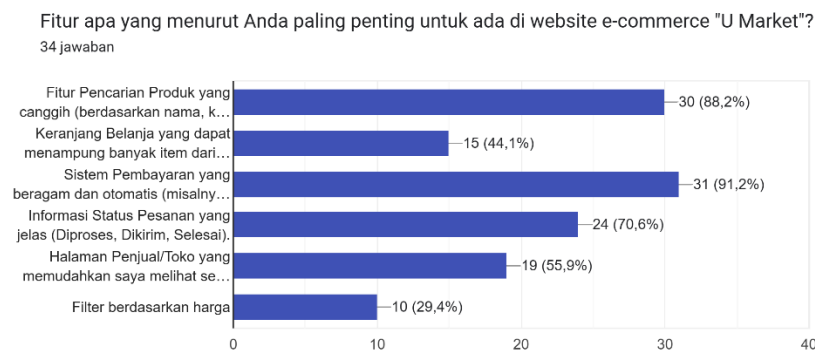
Apa kesulitan terbesar yang Anda hadapi saat berbelanja melalui grup WhatsApp "U Market"?

34 jawaban

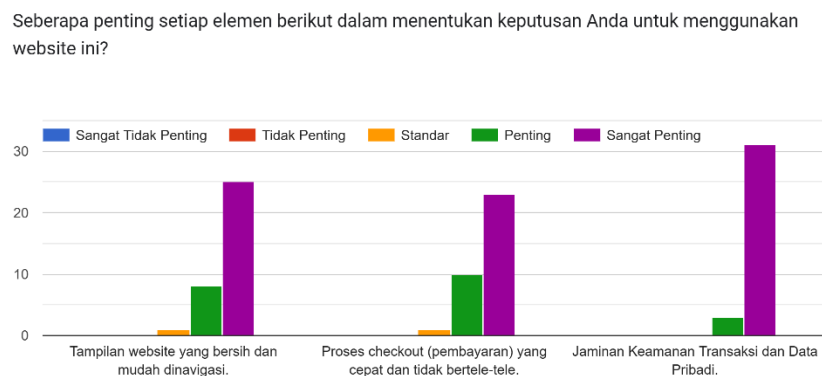
Produk yg ditawarkan sering tertumpuk
Penjual kurang responsif dalam komunikasi
Tidak lengkap keterangan produk
Penjual sangat banyak sehingga ketika ada yang menawarkan barang atau jasa banyak penjual lain pesannya tertimbun dengan pesan yang baru.
Chat suka tertimbun
Produknya sering tertimbun
Promosi produknya cepat tertimbun dan jadi spam, sehingga mencari produk nya sulit
Informasi produk sering tertimbun oleh chat lain, jadi sulit mencari kembali produk yang ingin dibeli. Selain itu, proses konfirmasi pesanan juga kadang lama karena harus menunggu respon dari penjual.

Gambar 3.3 Jawaban Responden Terkait Kesulitan Terbesar.

Berdasarkan diagram batang dan jawaban responden terkait kesulitan terbesar dalam melakukan aktivitas jual beli di WhatsApp U-Market, dapat diketahui bahwa permasalahan yang terjadi diantaranya informasi produk cepat tertimbun oleh pesan lain, sulit untuk mencari produk tertentu di dalam riwayat chat, konfirmasi pesanan dan pembayaran sering memakan waktu, serta 8 dari 34 responden pernah mengalami kesalahan dalam pemesanan karena komunikasi yang kurang baik melalui grup WhatsApp. Untuk menjawab permasalahan yang terjadi, responden diminta untuk memberikan pendapat terkait fitur serta tampilan seperti apa yang diharapkan.



Gambar 3.4 Jawaban Responden Terakit Fitur Paling Penting.



Gambar 3.5 Diagram Batang kepentingan Elemen.

Berdasarkan diagram pada Gambar 3.4 dan Gambar 3.5, dapat diketahui bahwa fitur yang paling diminati adalah sistem pembayaran yang beragam dan otomatis, pencarian produk yang canggih, dan informasi status pesanan yang jelas. Selain itu,

seluruh responden setuju dengan pentingnya beberapa elemen seperti tampilan *website* yang bersih dan mudah dinavigasi, proses *checkout* yang cepat dan ringkas, serta jaminan keamanan transaksi dan data pribadi. Setelah mengetahui permasalahan yang terjadi dan fitur yang diharapkan oleh responden, maka proses analisis mendalam terkait penentuan kebutuhan fungsional dan kebutuhan non fungsional dari calon pengguna sudah dapat dilakukan.

### 3. Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan layanan atau fitur yang harus disediakan oleh sistem agar dapat mendukung proses bisnis *e-commerce*. Adapun kebutuhan fungsional pada sistem *back-end* U-Market adalah sebagai berikut:

Tabel 3.4 Kebutuhan Fungsional

Kode Fungsional	Deskripsi
KF-01	Sistem harus mampu melakukan proses <i>login</i> dan <i>logout</i> pengguna dengan mekanisme autentikasi yang aman, serta membedakan hak akses antara admin dan pengguna berdasarkan peran ( <i>role</i> ) yang tersimpan di <i>database</i> .
KF-02	Sistem harus menyediakan layanan <i>back-end</i> yang memungkinkan pengguna (penjual) untuk menambahkan, memperbarui, menampilkan, dan menghapus data produk yang dijual, dengan data tersimpan secara konsisten pada <i>database</i> .
KF-03	Sistem harus menyediakan fitur pencarian produk melalui <i>endpoint back-end</i> berdasarkan nama produk atau kategori, sehingga memudahkan pengguna dalam menemukan produk yang diinginkan.
KF-04	Sistem harus mampu mencatat pesanan pengguna, memproses dan memperbarui status pembayaran, serta menyimpan riwayat transaksi pengguna secara terstruktur pada <i>database</i> .
KF-05	Sistem harus menyediakan fitur <i>back-end</i> yang memungkinkan admin melihat, memperbarui, serta menonaktifkan akun pengguna apabila diperlukan untuk keperluan pengelolaan sistem.
KF-06	Sistem harus menyediakan layanan <i>back-end</i> yang memungkinkan pengguna untuk melihat dan memperbarui data profil dengan menerapkan validasi <i>input</i> serta menyimpan perubahan data secara aman ke dalam <i>database</i> .

Tabel 3.4 Kebutuhan Fungsional (Lanjutan)

Kode Fungsional	Deskripsi
KF-07	Sistem <i>back-end</i> harus menyediakan <i>endpoint</i> REST API yang dapat diakses oleh <i>front-end</i> untuk menampilkan data produk, transaksi, dan data pengguna secara terstruktur dan aman.
KF-08	Sistem harus melakukan validasi input pada setiap proses ( <i>login</i> , pengelolaan produk, transaksi) untuk memastikan data yang tersimpan tetap konsisten, valid, dan terhindar dari kesalahan <i>input</i> .
KF-09	Sistem harus menyediakan fitur <i>back-end</i> yang memungkinkan admin melihat laporan penjualan dan data transaksi sebagai bahan monitoring dan evaluasi aktivitas <i>e-commerce</i> .

#### 4. Kebutuhan Non Fungsional

Kebutuhan non-fungsional menjelaskan aspek teknis yang harus dipenuhi agar sistem dapat berjalan secara optimal, aman, dan mudah digunakan. Adapun kebutuhan non fungsional pada sistem *back-end* U-Market adalah sebagai berikut:

Tabel 3.5 Kebutuhan Non Fungsional

Kode Non-Fungsional	Deskripsi
KNF-01	Sistem harus menerapkan mekanisme keamanan berupa enkripsi kata sandi pengguna dan token API, untuk melindungi data dari akses tidak sah.
KNF-02	Sistem harus mampu memberikan respon API yang cepat dan stabil, meskipun menangani banyak permintaan secara bersamaan, guna menjamin kenyamanan pengguna.
KNF-03	Sistem harus dapat beroperasi secara stabil dan andal tanpa sering mengalami gangguan, khususnya pada saat proses transaksi dan pengolahan data berlangsung.
KNF-04	Sistem harus dirancang dengan arsitektur yang mudah dikembangkan dan dipelihara, sehingga penambahan fitur di masa depan dapat dilakukan tanpa perubahan besar pada struktur utama sistem.
KNF-05	Kode program harus mengikuti pola arsitektur MVC pada <i>framework</i> Laravel agar struktur kode lebih terorganisasi dan memudahkan proses pengembangan serta pemeliharaan sistem.
KNF-06	Sistem harus dapat berjalan pada berbagai perangkat dan terintegrasi dengan antarmuka pengguna berbasis web melalui layanan REST API.

### 3.3.2 Product Backlog

Pada tahap *Product Backlog*, dilakukan penyusunan daftar seluruh fitur, fungsionalitas, dan kebutuhan produk yang akan dikembangkan. *Product Owner* bersama tim pengembang bersinergi dalam menentukan prioritas serta estimasi waktu pada *Product Backlog*. Dalam penentuan skala prioritas, *item* dengan *story point* yang lebih besar diprioritaskan untuk dikerjakan lebih dulu karena tingkat kompleksitas dan durasi pengerjaannya yang tinggi. Fokus utama tim adalah penyelesaian fitur inti, sedangkan fitur tambahan baru akan dipertimbangkan pada tahap pengembangan selanjutnya jika tujuan utama telah tercapai. Adapun standar estimasi pengerjaan yang digunakan mengacu pada *sizing rule* berikut:

- a. 0.5 *Story point*: < 4 jam (tugas ringan).
- b. 1 *Story point*: 4 jam (setengah hari jam kerja).
- c. 2 *Story points*: 8 jam (1 hari jam kerja).
- d. 3 *Story points*: 16 jam (2 hari jam kerja).
- e. 5 *Story points*: 24–32 jam (3-4 hari jam kerja).
- f. 8 *Story points*: 40 jam (satu minggu atau separuh durasi *Sprint*).

Berikut merupakan *Product Backlog* untuk *website* U-Market.

Tabel 3.6 *Product Backlog* Pengguna (Fitur Utama)

No.	Item	Prioritas	Points
1	Inisialisasi proyek <i>backend</i> Laravel, konfigurasi struktur folder dan <i>routing</i> REST API	Sedang	2
2	Konfigurasi koneksi <i>database</i> MySQL dan <i>environment backend</i>	Rendah	0.5
3	Pemodelan <i>database website</i> U-Market	Rendah	1
4	Implementasi API autentikasi ( <i>register, login, logout</i> )	Tinggi	5
5	<i>Hash Password</i>	Rendah	1
6	Implementasi <i>Role Based Access Control</i> (RBAC) pengguna	Sedang	2
7	Implementasi API manajemen profil pengguna (lihat & edit profil)	Sedang	3
8	Implementasi API daftar produk	Tinggi	3
9	Implementasi API detail/deskripsi produk	Tinggi	3
10	Implementasi API pencarian produk (nama/kategori)	Sedang	2
11	Implementasi API kelola keranjang pengguna	Tinggi	3

Tabel 3.6 *Product Backlog* Pengguna (Fitur Utama) (Lanjutan)

No.	Item	Prioritas	Points
12	Implementasi API <i>checkout</i> dengan integrasi <i>Payment gateway</i> Midtrans	Tinggi	5
13	Implementasi API <i>callback</i> Midtrans untuk validasi dan <i>update</i> status transaksi	Tinggi	5
14	Implementasi API akses halaman penjual	Sedang	3
15	Implementasi API riwayat transaksi pengguna	Sedang	3
16	Implementasi Integrasi Kontak Penjual (WhatsApp)	Sedang	2
17	Implementasi API verifikasi email pengguna	Tinggi	5
18	Implementasi API manajemen produk penjual	Tinggi	5
19	Implementasi API manajemen rekening bank	Tinggi	5
20	Implementasi API manajemen saldo dan penarikan dana	Tinggi	5

Tabel 3.7 *Product Backlog* Pengguna (Fitur Tambahan)

No.	Item	Prioritas	Points
1	Penerapan enkripsi <i>password</i> dan token autentikasi	Tinggi	3
2	Validasi input pada seluruh <i>endpoint</i> pengguna	Sedang	2
3	Optimasi <i>query database</i> untuk performa API	Sedang	2
4	Implementasi <i>middleware</i> autentikasi API	Tinggi	3
5	Penanganan <i>error</i> dan <i>response</i> API terstandarisasi	Sedang	2
6	<i>Logging</i> aktivitas transaksi dan pembayaran	Sedang	2
7	Pengujian REST API menggunakan Postman	Sedang	3
8	Pengujian <i>Black box</i> fungsional sistem	Sedang	3

Tabel 3.8 *Product Backlog* Admin (Fitur Utama)

No.	Item	Prioritas	Points
1	Implementasi API <i>login</i> dan autentikasi admin	Tinggi	3
2	Implementasi API beranda Admin	Tinggi	5
3	Implementasi API manajemen pengguna	Tinggi	5
4	Implementasi API manajemen produk	Tinggi	5
5	Implementasi API melihat seluruh transaksi	Tinggi	3
6	Implementasi pembatasan akses <i>endpoint</i> admin	Tinggi	3

Tabel 3.9 *Product Backlog* Admin (Fitur Tambahan)

No.	Item	Prioritas	Points
1	Penerapan otorisasi berbasis <i>role</i> admin	Tinggi	3
2	Validasi dan sanitasi input pada <i>endpoint</i> admin	Sedang	2
3	<i>Logging</i> aktivitas admin	Rendah	1
4	Dokumentasi REST API admin	Rendah	0.5
5	<i>Deployment backend</i> ke <i>server</i> produksi	Sedang	2
6	Pengujian REST API <i>Black box Testing</i> Admin	Sedang	3

Selanjutnya terdapat desain sistem yang menggambarkan fitur serta fungsionalitas dari *website* yang dibangun. Pada penelitian ini akan digambarkan menggunakan diagram UML diantaranya, *use case diagram*, *activity diagram*, dan *class diagram*.

### 1. *Use case diagram*

Perancangan *use case diagram* membantu dalam proses pengembangan sistem karena memberikan gambaran awal mengenai modul-modul yang harus diimplementasikan. Setiap *use case* yang diidentifikasi kemudian menjadi acuan dalam penyusunan *Sprint* pengembangan serta perancangan struktur basis data dan alur proses sistem. Dengan demikian, *use case diagram* tidak hanya berfungsi sebagai dokumentasi, tetapi juga sebagai pedoman teknis dalam tahap implementasi.

*Use case diagram* menggambarkan interaksi antara pengguna (aktor) dengan sistem yang dikembangkan. Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh masing-masing aktor terhadap sistem *back-end e-commerce* U-Market. Sistem U-Market memiliki dua aktor utama, yaitu Admin dan Pengguna. Pengguna dalam sistem ini mencakup peran sebagai pembeli maupun penjual. Berikut merupakan tabel yang menyajikan deskripsi dari masing-masing aktor:

Tabel 3.10 Deskripsi Aktor

No	Aktor	Deskripsi
1	Admin	Pihak yang memiliki hak akses penuh dalam sistem U-Market untuk mengelola data produk, pengguna, dan transaksi, serta melakukan pengawasan terhadap seluruh aktivitas yang terjadi pada sistem.
2	Pengguna	<ul style="list-style-type: none"> <li>• Pengunjung Pihak yang mengakses sistem U-Market tanpa melakukan proses autentikasi. Pengunjung hanya dapat melihat informasi umum dan registrasi.</li> <li>• Pengguna Terdaftar Pihak yang telah melakukan registrasi dan login ke dalam sistem U-Market sehingga memiliki hak akses lebih luas. Pengguna terdaftar dapat mengelola profil, menambahkan produk ke keranjang, melakukan transaksi pembelian, melihat riwayat transaksi, dan mengakses halaman toko saya.</li> </ul>

Melalui identifikasi aktor dan interaksi yang terjadi, peneliti dapat menentukan batasan sistem (*system boundary*) serta hubungan antara pengguna dan fitur yang tersedia. Hal ini penting untuk memastikan bahwa setiap kebutuhan yang telah dianalisis pada tahap sebelumnya dapat direpresentasikan secara jelas dalam bentuk visual. Gambaran dari *use case diagram website e-commerce* U-Market disajikan pada Gambar 3.6. *Use case* yang telah digambarkan selanjutnya didefinisikan lebih lanjut dalam tabel 3.11.



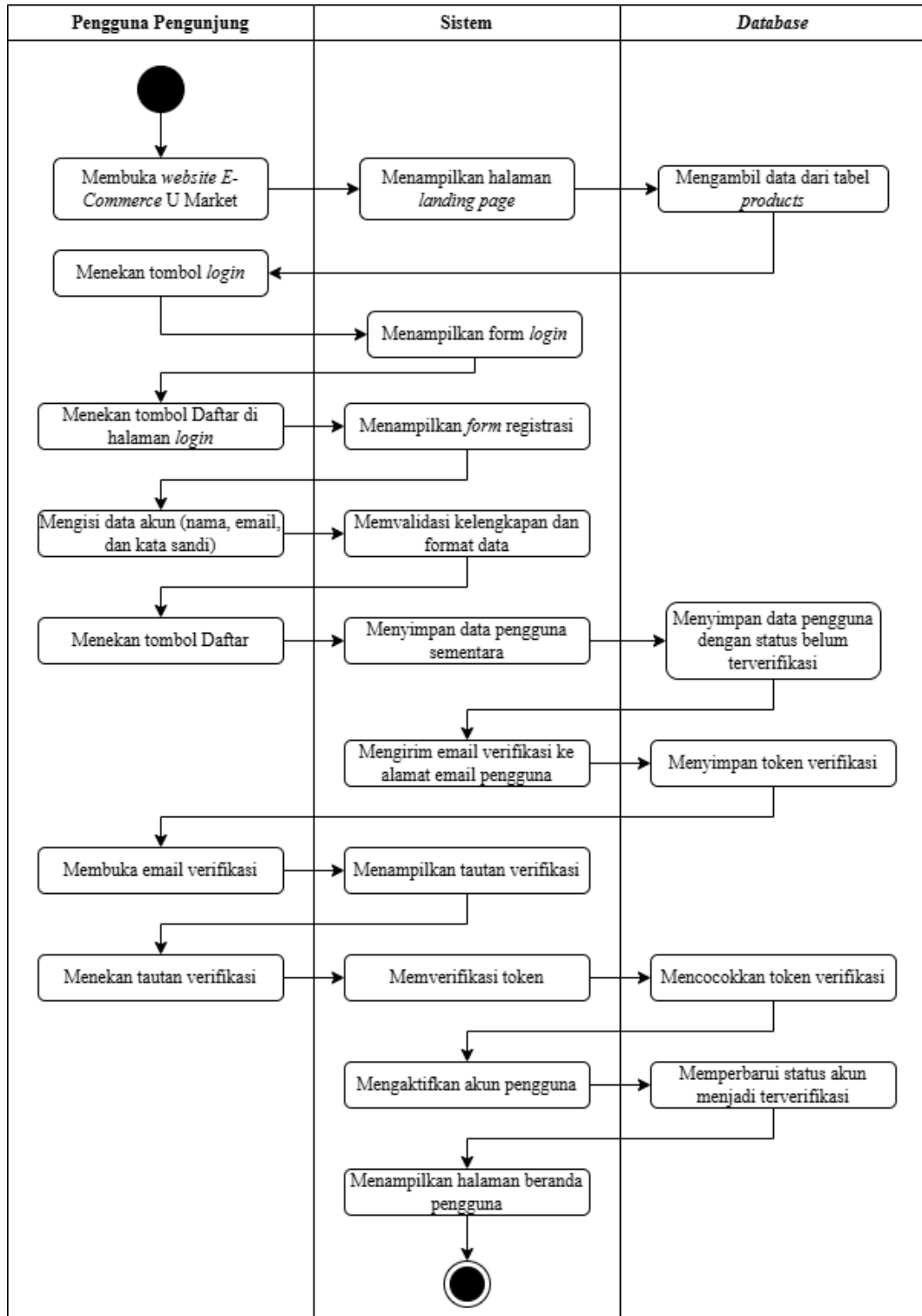
Gambar 3.6 Use Case Diagram U-Market

Tabel 3.11 Definisi *Use case*

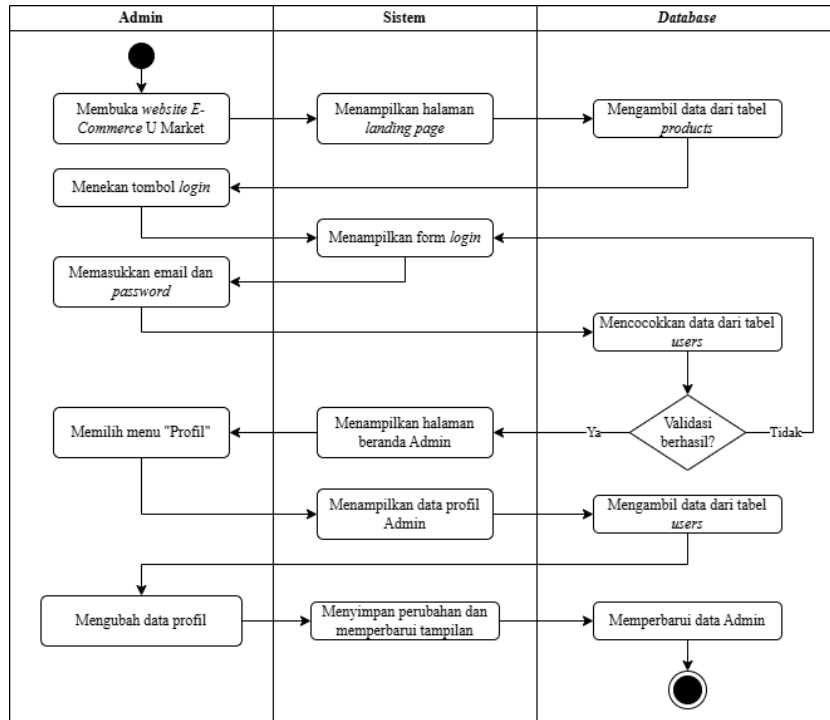
No	Aktor	<i>Use case</i>	Deskripsi
1	Admin	Mengelola Seluruh Produk	Mengelola data produk yang dijual oleh penjual, meliputi pemantauan dan pengawasan produk pada sistem.
2	Admin	Mengelola Pengguna	Mengelola data pengguna yang terdaftar pada sistem, termasuk melihat dan menonaktifkan akun pengguna.
3	Admin	Mengelola Transaksi	Melihat dan memantau data transaksi yang terjadi pada sistem U-Market.
4	Admin	Mengelola Profil Admin	Mengubah serta memperbarui data profil admin.
5	Admin, Pengguna	<i>Logout</i>	Berhenti mengakses sistem dan keluar dari sistem.
6	Pengguna Terdaftar	Mengelola Profil	Mengubah dan memperbarui data profil pengguna.
7	Pengguna Terdaftar	Mengelola Keranjang	Menambahkan, mengubah jumlah, dan menghapus produk di dalam keranjang belanja.
8	Pengguna Terdaftar	Mengakses Status Pesanan	Melihat dan memantau status pesanan yang telah dilakukan.
9	Pengguna Terdaftar	Melakukan Transaksi	Melakukan proses pemesanan dan pembayaran produk.
10	Pengguna Terdaftar	Mengakses Halaman Toko Saya	Mengelola produk yang dijual, meliputi menambah, mengubah, dan menghapus produk.
11	Pengunjung	Mengakses Informasi Produk	Melihat dan menelusuri informasi produk yang tersedia pada sistem.
12	Pengunjung	Registrasi	Mendaftarkan akun agar dapat mengakses sistem.
13	Admin, Pengguna	<i>Login</i>	Melakukan autentikasi untuk mengakses sistem sesuai dengan hak akses masing-masing pengguna.

## 2. Activity diagram

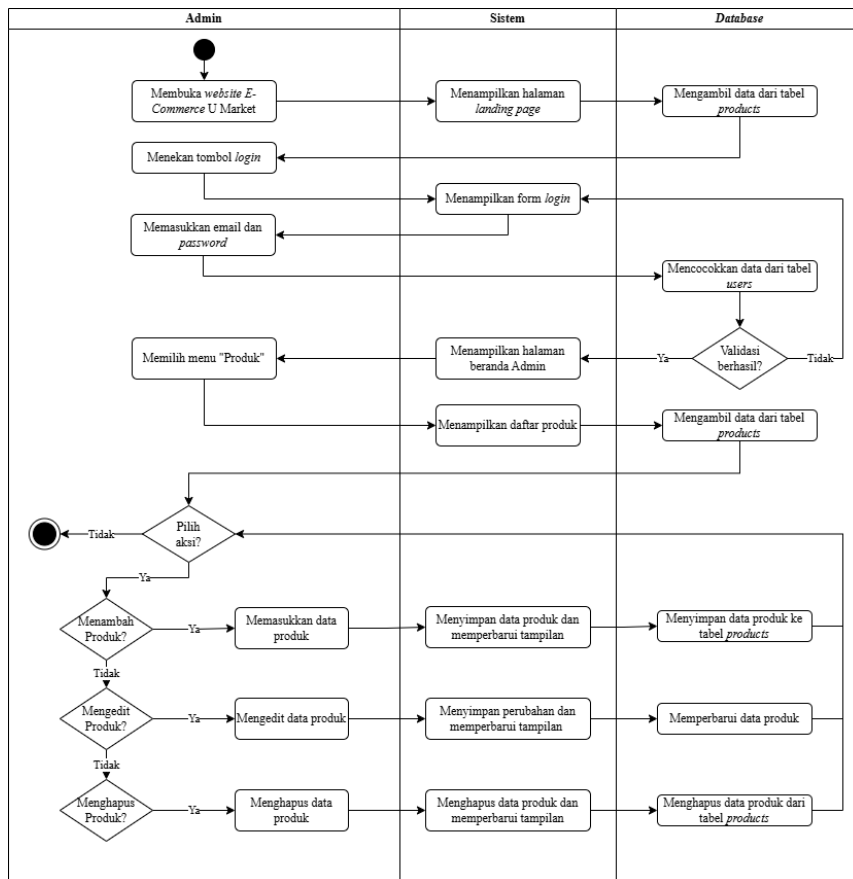
*Activity diagram* digunakan untuk memodelkan alur aktivitas yang terjadi dalam sistem, menggambarkan bagaimana proses bisnis berjalan dari awal hingga akhir. Diagram ini juga memperlihatkan interaksi antar pengguna, sistem, dan *database* secara kronologis. Berikut merupakan aktivitas utama dalam sistem *back-end e-commerce* U-Market.



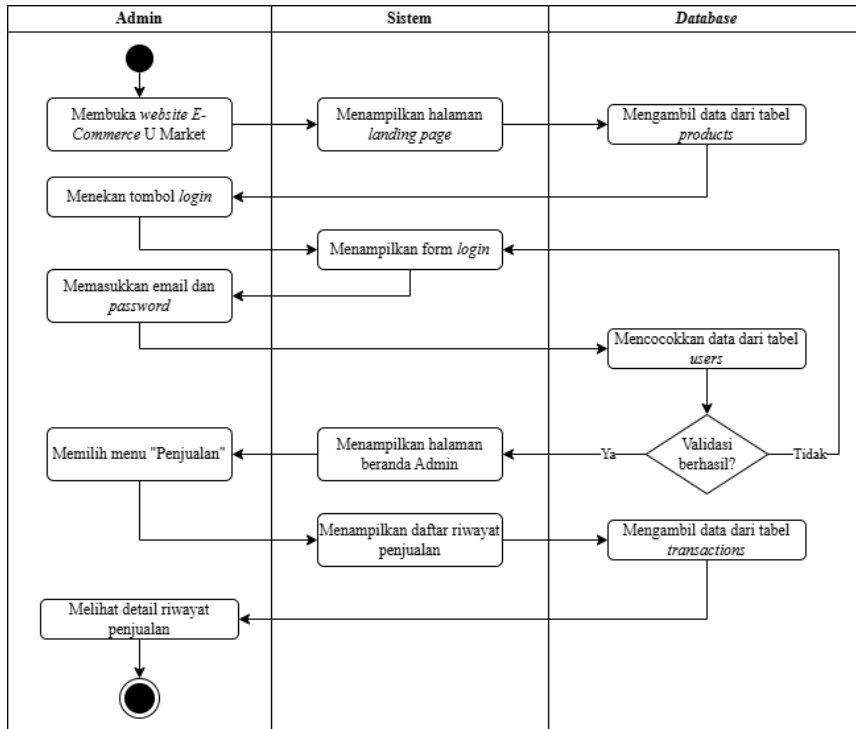
Gambar 3.7 Activity Diagram Registrasi



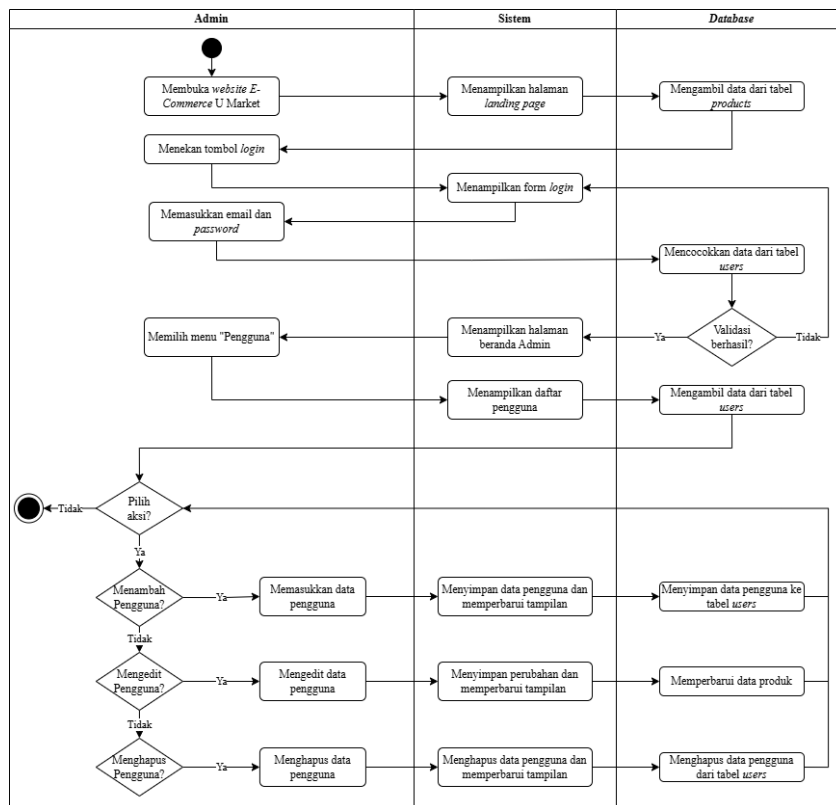
Gambar 3.8 Activity Diagram Mengelola Profil Admin



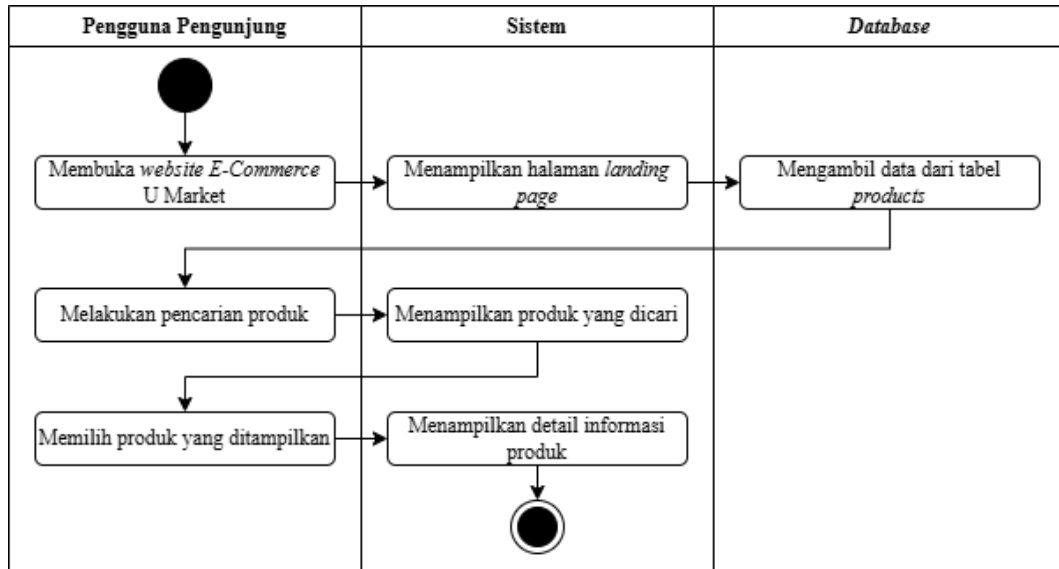
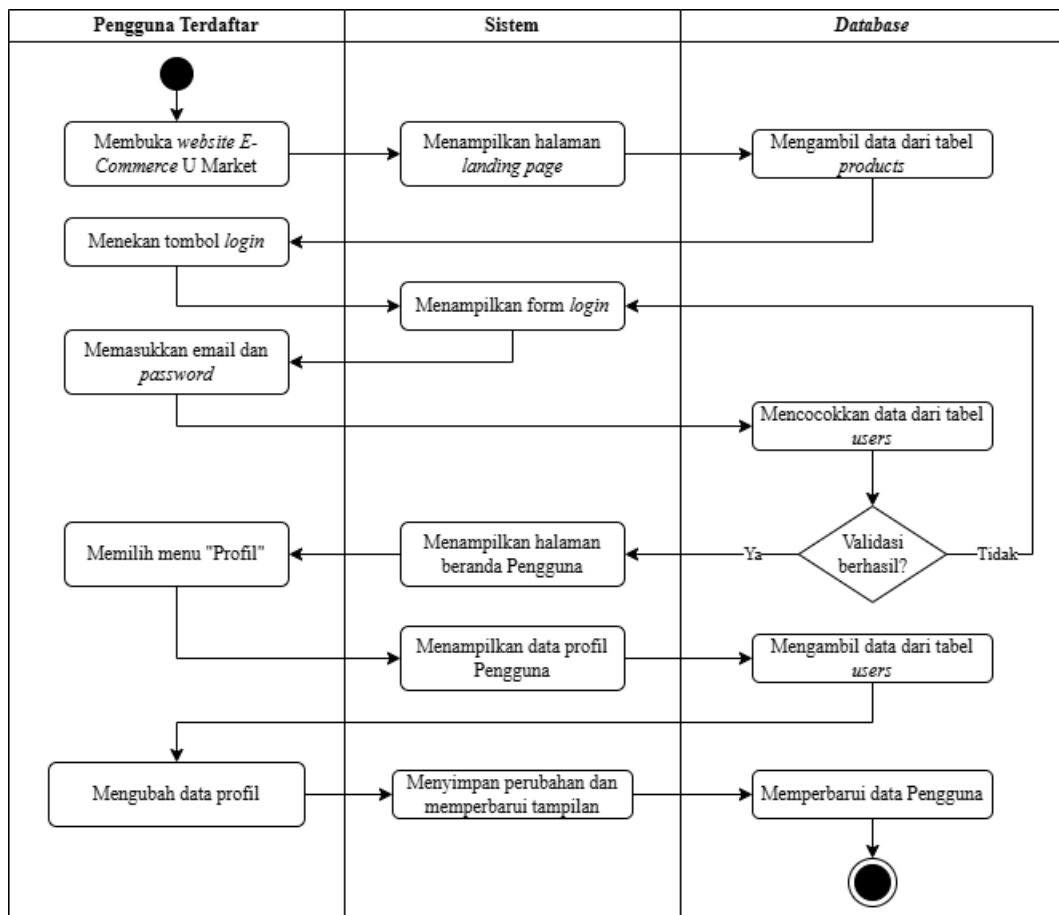
Gambar 3.9 Activity Diagram Mengelola Seluruh Produk

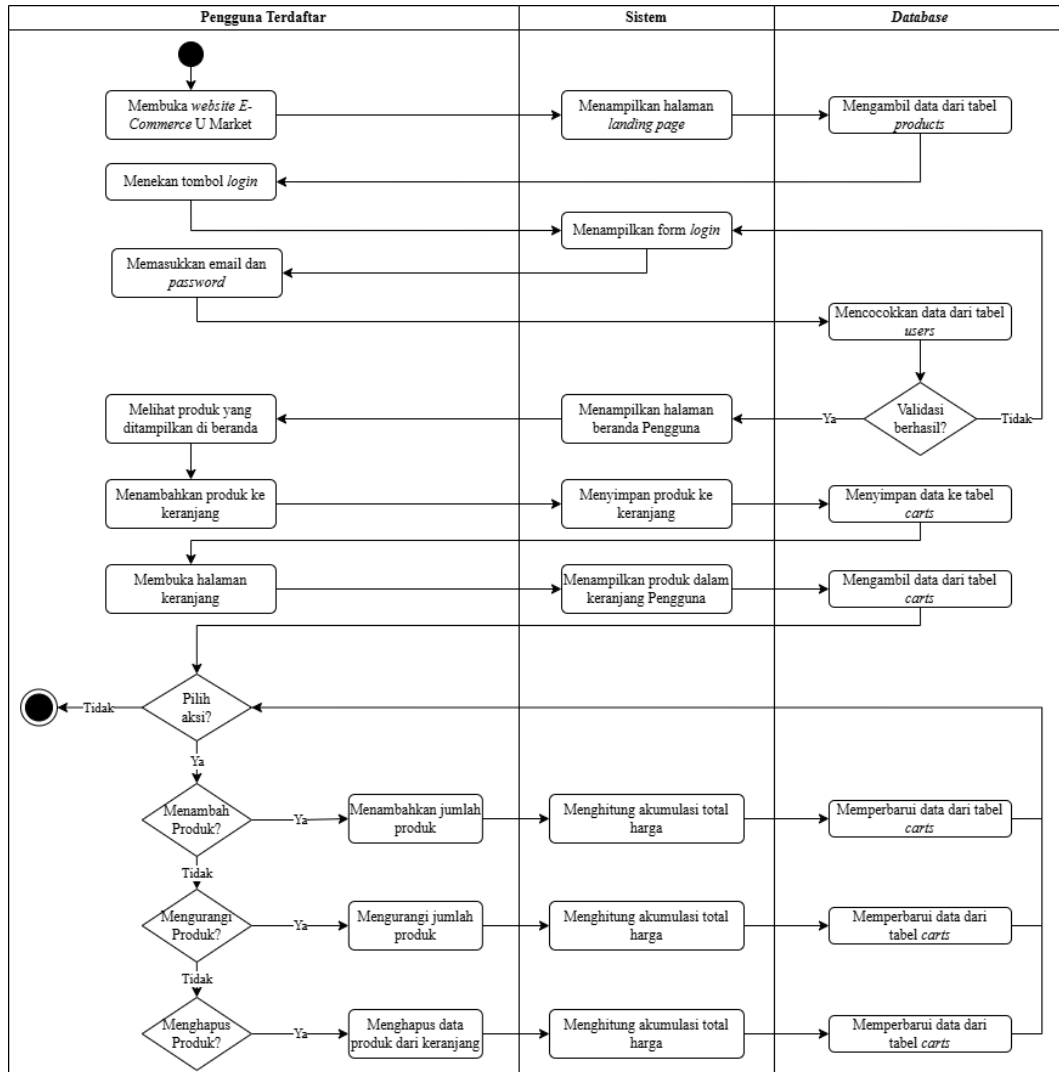


Gambar 3.10 Activity Diagram Mengelola Transaksi

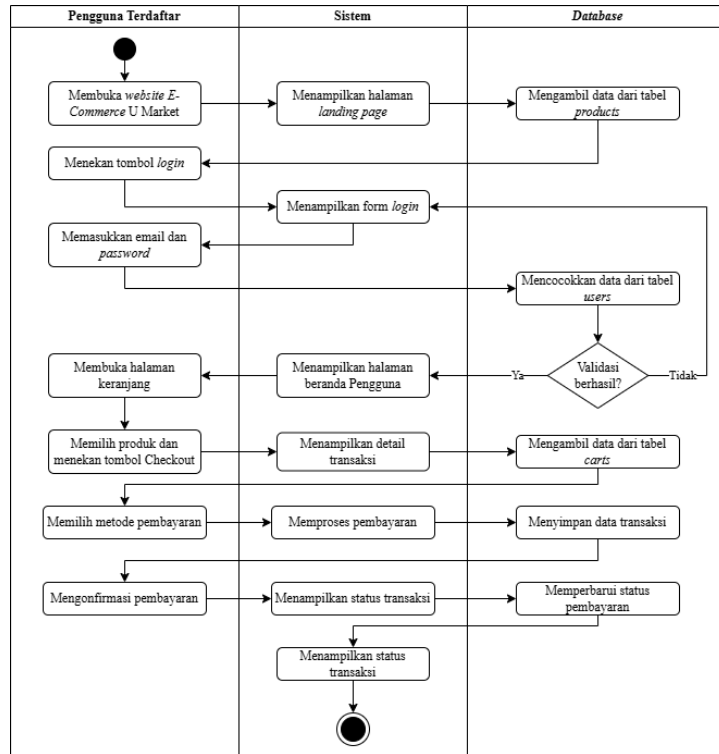


Gambar 3.11 Activity Diagram Mengelola Pengguna

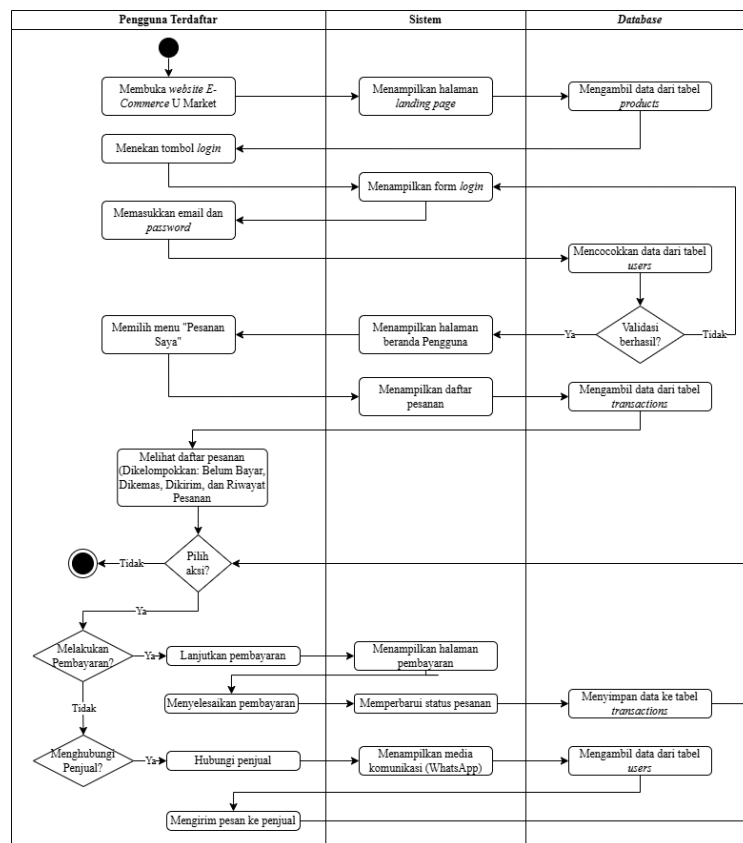
Gambar 3.12 *Activity Diagram* Mengakses Informasi ProdukGambar 3.13 *Activity Diagram* Mengelola Profil Pengguna



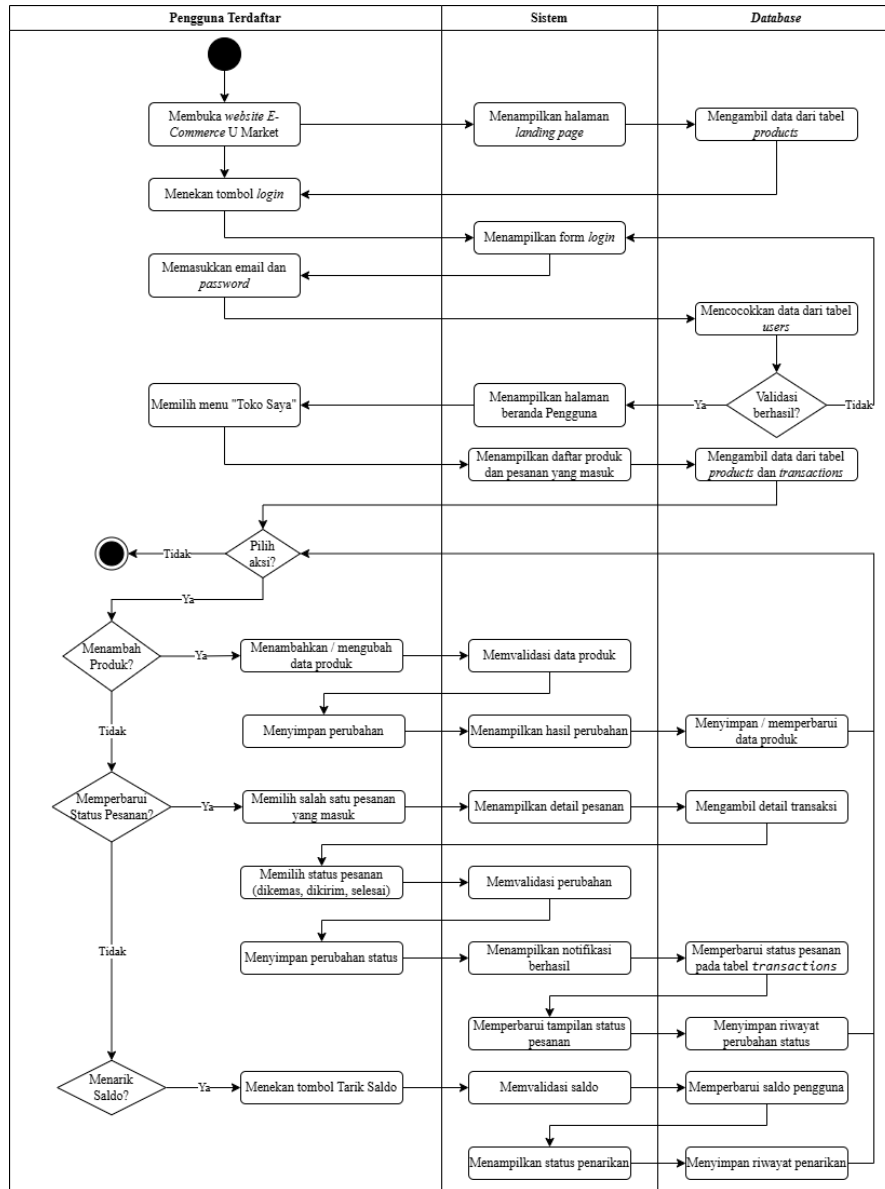
Gambar 3.14 Activity Diagram Mengelola Keranjang



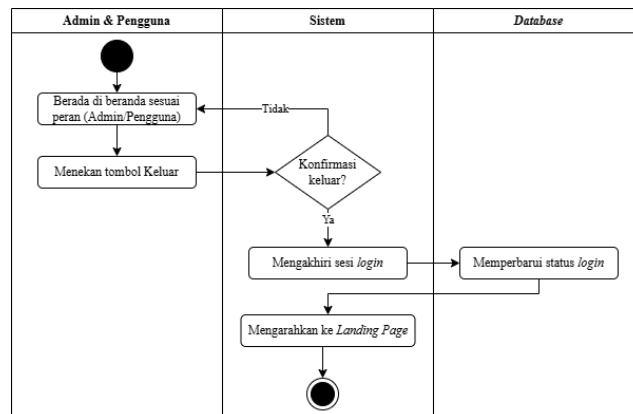
Gambar 3.15 Activity Diagram Melakukan Transaksi



Gambar 3.16 Activity Diagram Mengakses Status Pesanan



Gambar 3.17 Activity Diagram Mengkses Halaman Toko Saya



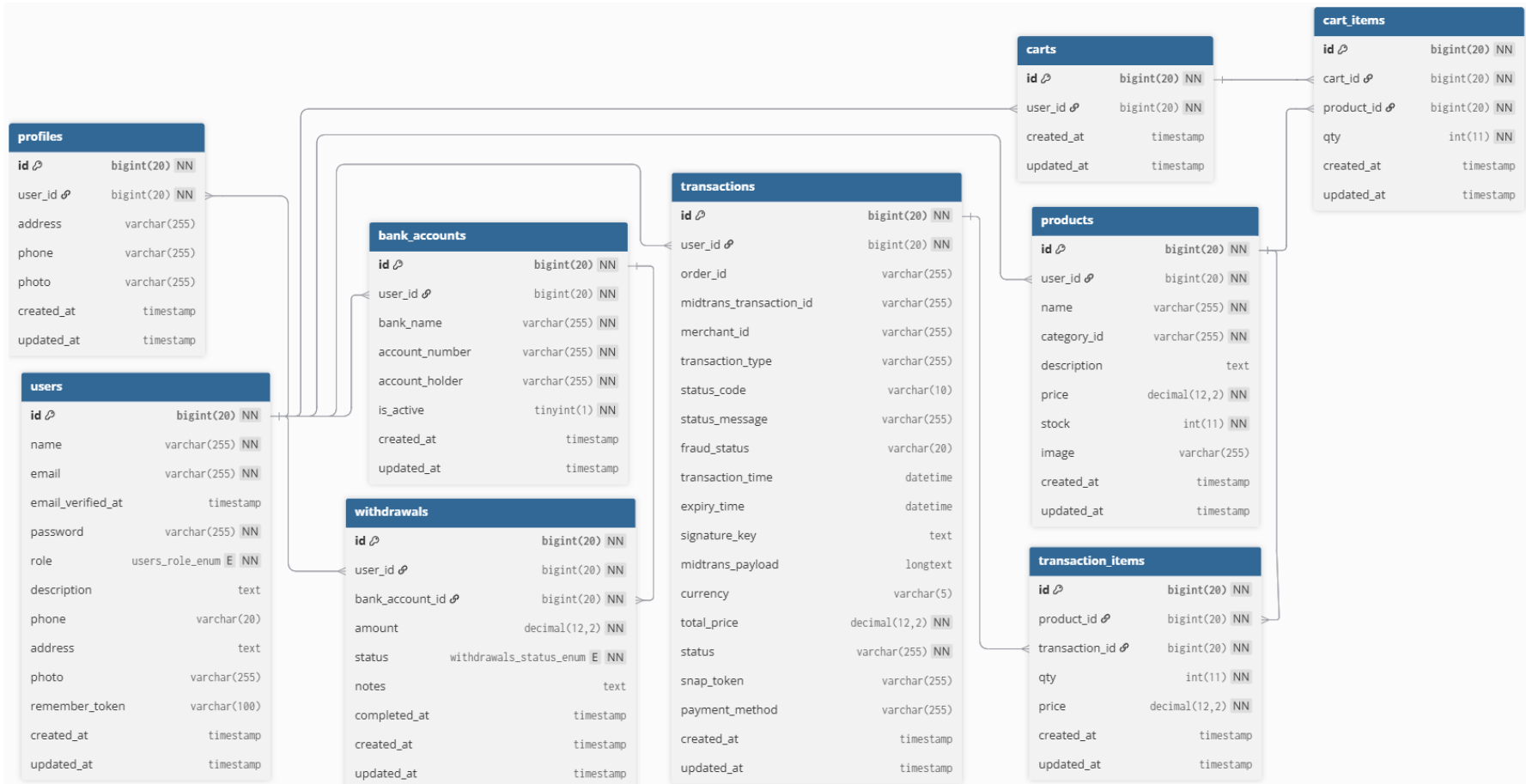
Gambar 3.18 Activity Diagram Logout

### 3. *Class Diagram*

*Class diagram* menggambarkan struktur sistem secara statis, termasuk kelas, atribut, operasi (*method*), serta hubungan antar kelas dalam sistem. Diagram ini membantu pengembang memahami arsitektur sistem *back-end* yang akan diimplementasikan menggunakan Laravel.

Pada *class diagram* U-Market, entitas utama dalam sistem adalah *users* yang berperan sebagai pusat aktivitas. Setiap pengguna memiliki relasi satu ke satu dengan tabel *profiles*, yang menyimpan informasi profil. Pemisahan ini bertujuan untuk menjaga modularitas struktur data serta mempermudah pengelolaan informasi pengguna. Selanjutnya pengguna memiliki relasi satu ke banyak dengan tabel *Products*, yang merepresentasikan produk yang dijual oleh pengguna (penjual). Relasi ini menunjukkan bahwa satu pengguna dapat memiliki banyak produk, namun satu produk hanya dimiliki oleh satu pengguna. Dalam proses pembelian, sistem menggunakan entitas *carts* dan *cart\_items*. Relasi antara *carts* dan *cart\_items* bersifat satu ke banyak, sementara *cart\_items* memiliki relasi ke *Products* untuk mengidentifikasi produk yang dipilih. Proses transaksi direpresentasikan melalui tabel *transactions* dan *transaction\_items*. Setiap transaksi memiliki relasi satu ke banyak dengan *transaction\_items*, yang menyimpan rincian produk yang dibeli dalam transaksi tersebut, termasuk jumlah dan harga pada saat pembelian. Relasi ini memungkinkan satu transaksi terdiri dari beberapa produk sekaligus. Selain fitur transaksi, sistem juga mendukung fitur pengelolaan saldo penjual melalui tabel *bank\_accounts* dan *withdrawals*. Relasi antara *users* dan *bank\_accounts* serta *users* dan *withdrawals* bersifat satu ke banyak, yang menunjukkan bahwa satu pengguna dapat memiliki lebih dari satu rekening bank dan dapat melakukan beberapa kali penarikan saldo.

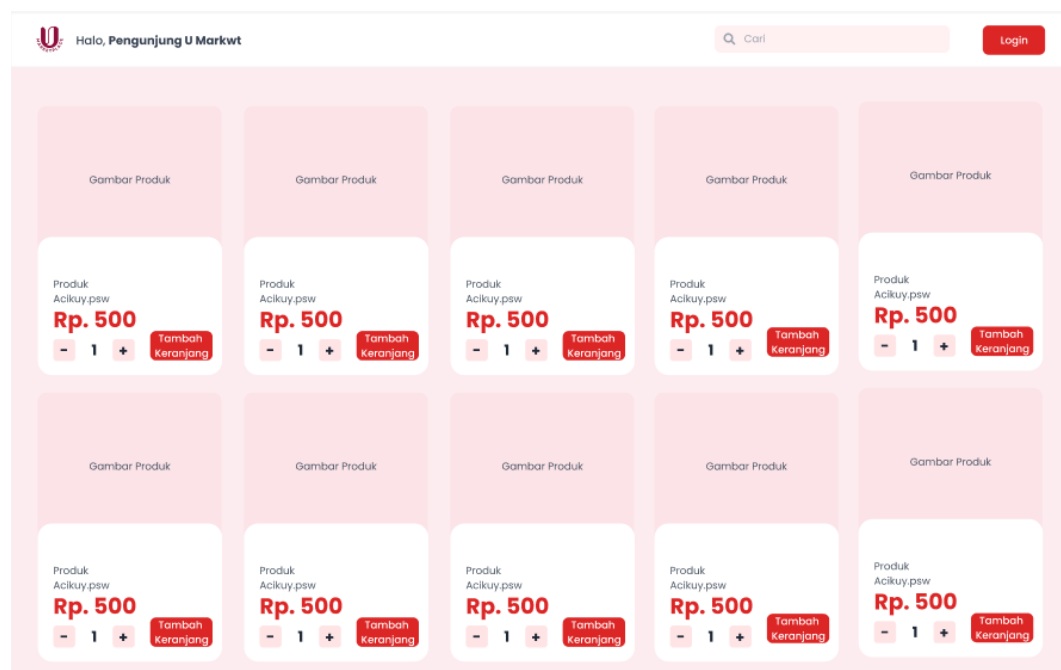
Secara keseluruhan, *class diagram* ini menggambarkan hubungan relasional yang terstruktur antara entitas pengguna, produk, keranjang, transaksi, serta sistem penarikan saldo. *Class diagram* dari sistem *back-end e-commerce* U-Market disajikan pada gambar 3.19.



Gambar 3.19 Class Diagram U-Market

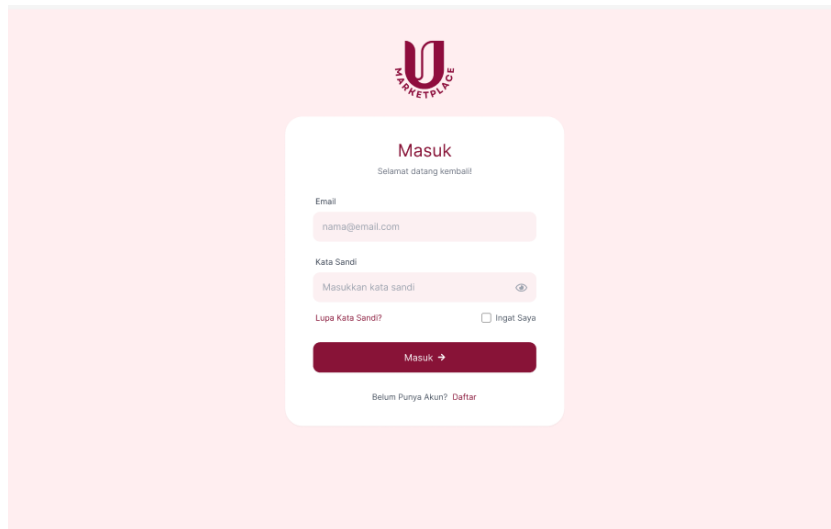
#### 4. *Mockup*

Perancangan antarmuka merupakan tahap yang berfokus pada perwujudan tampilan setiap halaman yang akan ditampilkan pada aplikasi. Tahap desain antarmuka ini dilakukan setelah perancangan sistem selesai, dengan tujuan untuk memastikan kesesuaian antara kebutuhan sistem dan tampilan yang dihasilkan. Perancangan antarmuka dilakukan secara terstruktur dan ditetapkan sejak awal proses pengembangan guna menghindari terjadinya pengulangan penulisan kode. Pada penelitian ini, perancangan antarmuka dilakukan menggunakan perangkat lunak Figma yang bersifat *open-source*. Berikut ditampilkan *mockup* antarmuka *website* U-Market yang akan dikembangkan.



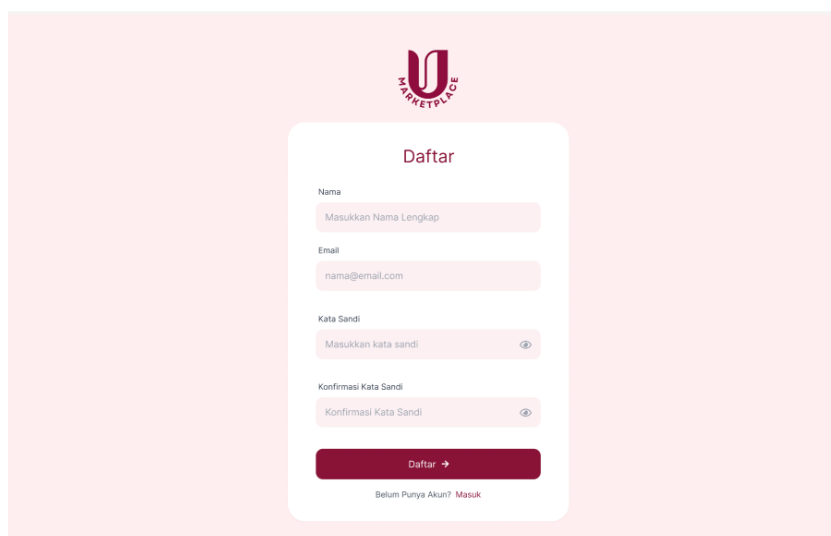
Gambar 3.20 *Mockup Landing page*

Gambar 3.20 menampilkan halaman utama *website* U-Market yang dapat diakses oleh pengguna tanpa perlu *login*. Pada halaman ini ditampilkan daftar produk dalam bentuk *grid* yang berisi gambar produk, nama produk, harga, serta tombol aksi untuk menambahkan produk ke keranjang. *Landing page* berfungsi sebagai media awal untuk memperkenalkan produk dan memudahkan pengguna dalam melakukan eksplorasi produk.



Gambar 3.21 *Mockup* Halaman *Login*

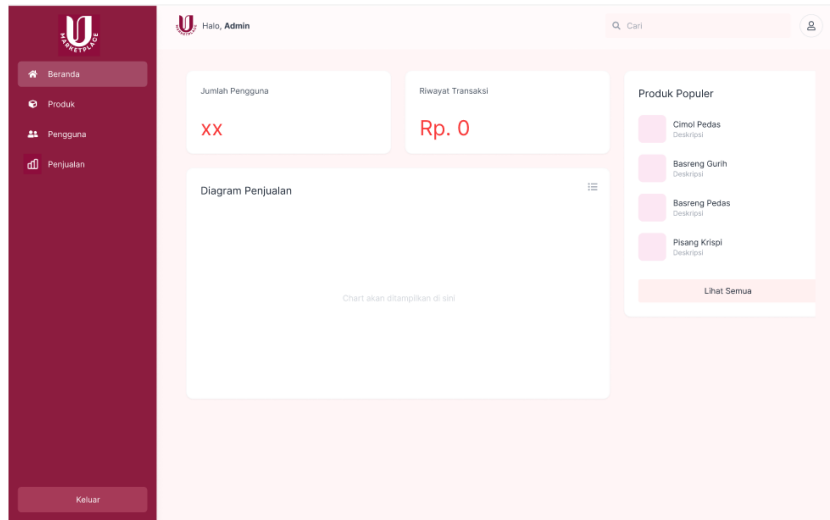
Gambar 3.21 menampilkan halaman *login* yang digunakan sebagai antarmuka autentikasi pengguna untuk masuk ke dalam sistem. Halaman ini menyediakan *input* email dan kata sandi yang digunakan oleh seluruh jenis pengguna. Sistem akan memvalidasi data *login* dan mengarahkan pengguna ke beranda yang sesuai dengan peran akun yang terdaftar.



Gambar 3.22 *Mockup* Halaman Registrasi

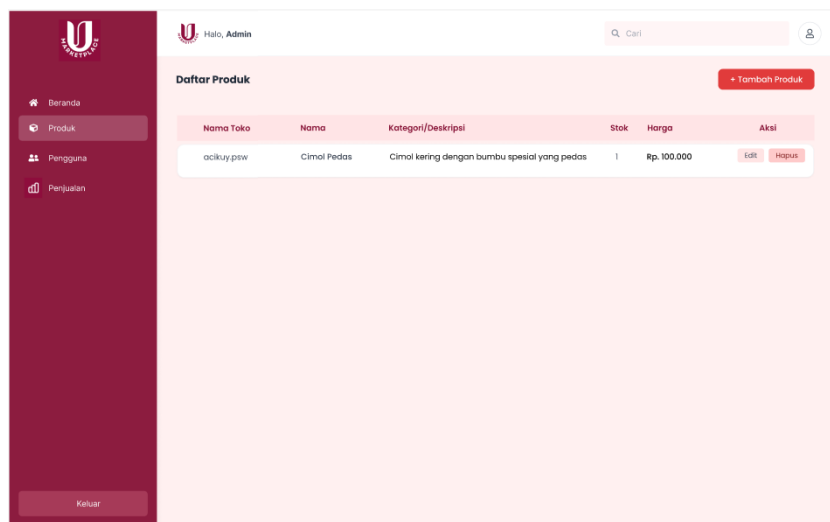
Gambar 3.22 menampilkan halaman registrasi yang berfungsi sebagai antarmuka pendaftaran akun baru bagi pengguna. Halaman ini menyediakan form pengisian

data berupa nama lengkap, email, kata sandi, dan konfirmasi kata sandi. Registrasi dilakukan untuk memberikan akses pengguna agar dapat melakukan transaksi dan menggunakan fitur yang tersedia pada sistem.



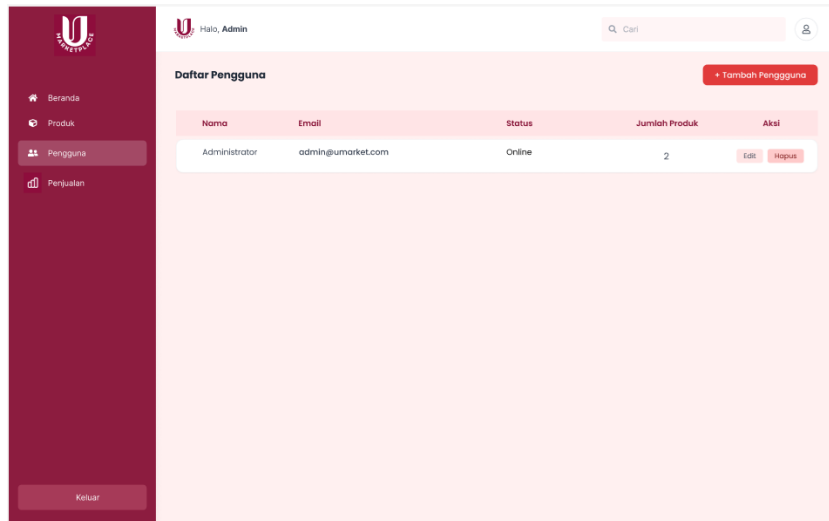
Gambar 3.23 *Mockup* Halaman Beranda Admin

Gambar 3.23 menampilkan halaman beranda admin yang berfungsi sebagai halaman utama setelah admin berhasil *login*. Halaman ini menampilkan ringkasan informasi sistem berupa jumlah pengguna, riwayat transaksi, produk populer, serta grafik penjualan untuk membantu admin memantau kondisi dan aktivitas sistem secara keseluruhan.



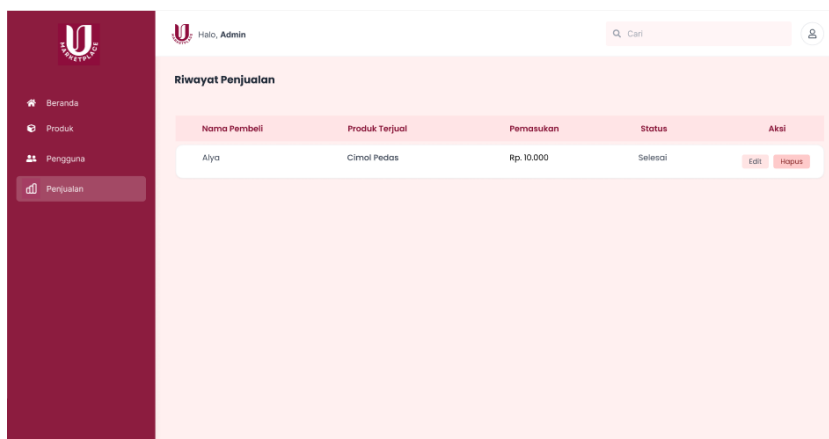
Gambar 3.24 *Mockup* Halaman Produk Admin

Gambar 3.24 menampilkan halaman produk admin yang digunakan untuk mengelola data produk yang tersedia pada sistem. Halaman ini menampilkan daftar produk beserta informasi nama toko, kategori, stok, dan harga, serta menyediakan fitur tambah, ubah, dan hapus produk untuk memudahkan pengelolaan data.



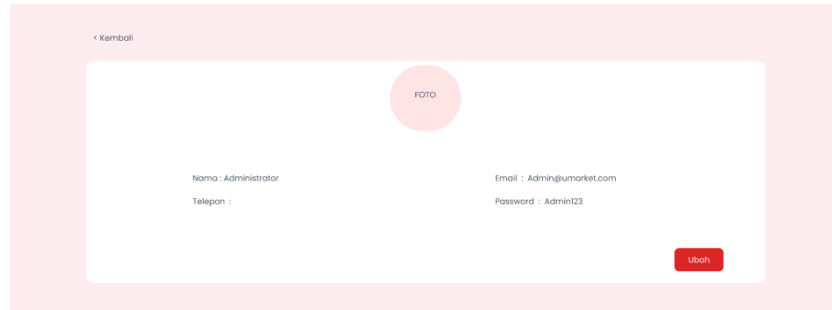
Gambar 3.25 *Mockup* Halaman Pengguna Admin

Gambar 3.25 menampilkan halaman pengguna admin yang berfungsi untuk mengelola data pengguna yang terdaftar pada sistem. Pada halaman ini ditampilkan informasi pengguna seperti nama, email, status akun, dan jumlah produk, serta menyediakan aksi pengelolaan data pengguna oleh admin.



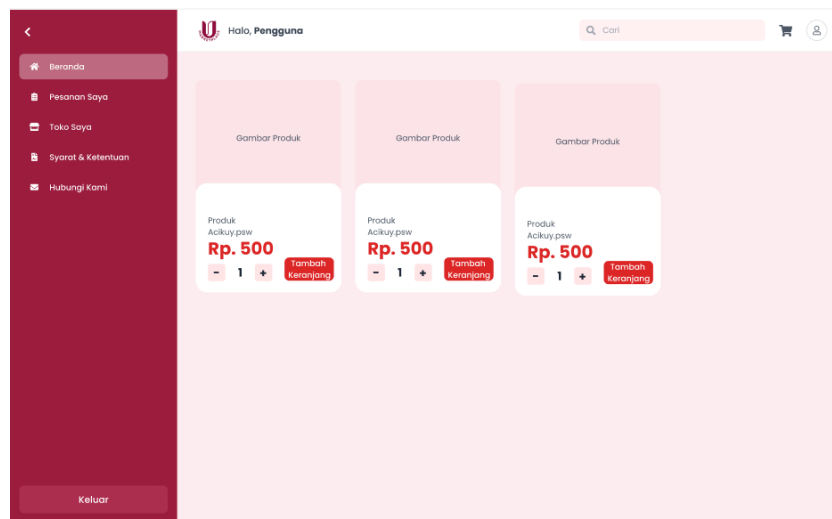
Gambar 3.26 *Mockup* Halaman Penjualan Admin

Gambar 3.26 menampilkan halaman penjualan admin yang berfungsi untuk menampilkan data riwayat transaksi penjualan pada sistem. Halaman ini memuat informasi nama pembeli, produk terjual, total pemasukan, dan status transaksi, serta menyediakan aksi pengelolaan data penjualan oleh admin.



Gambar 3.27 *Mockup* Halaman Edit Profil Admin

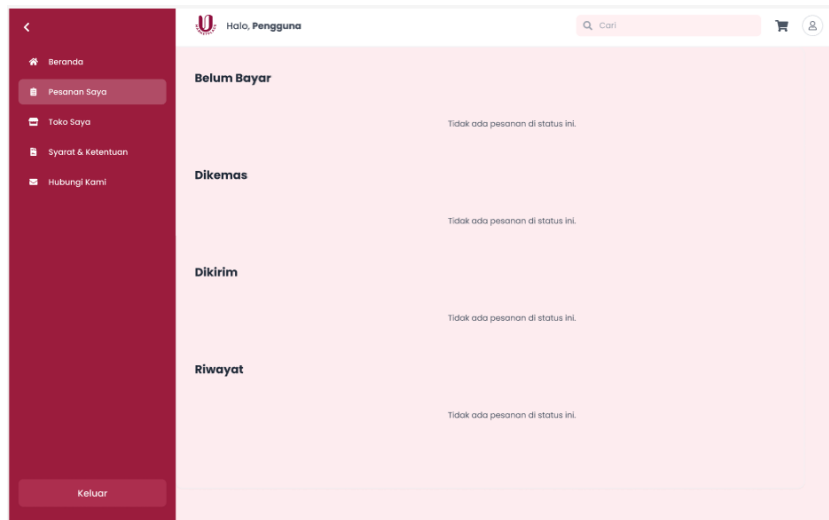
Gambar 3.27 menampilkan halaman edit profil admin yang digunakan untuk mengelola dan memperbarui data profil admin. Halaman ini menampilkan informasi akun seperti nama, email, dan kata sandi yang dapat diubah guna menjaga keakuratan dan keamanan data akun admin.



Gambar 3.28 *Mockup* Halaman Beranda Pengguna

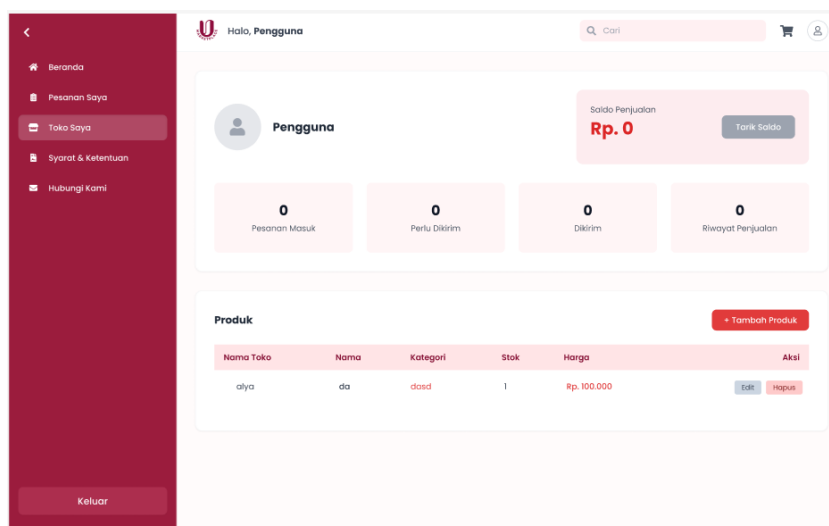
Gambar 3.28 menampilkan halaman beranda pengguna yang berfungsi sebagai halaman utama setelah pengguna berhasil *login*. Halaman ini menampilkan daftar produk yang tersedia lengkap dengan informasi harga dan tombol aksi untuk

menambahkan produk ke keranjang, sehingga memudahkan pengguna dalam melakukan transaksi.



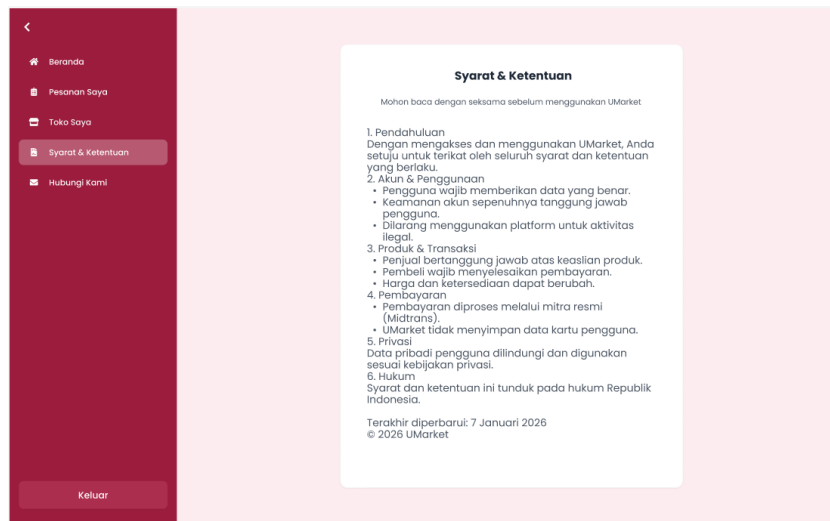
Gambar 3.29 *Mockup* Halaman Pesanan Saya

Gambar 3.29 menampilkan halaman pesanan saya yang berfungsi untuk menampilkan informasi status pesanan yang dilakukan oleh pengguna. Halaman ini mengelompokkan pesanan berdasarkan status seperti belum bayar, dikemas, dikirim, dan riwayat, sehingga memudahkan pengguna dalam memantau proses transaksi.



Gambar 3.30 *Mockup* Halaman Toko Saya

Gambar 3.30 menampilkan halaman buka toko yang digunakan sebagai antarmuka bagi pengguna yang ingin mendaftarkan toko pada sistem. Halaman ini menampilkan informasi profil pengguna, ringkasan aktivitas, serta form dan tabel pengelolaan produk yang memungkinkan pengguna menambahkan dan mengelola produk yang dijual.



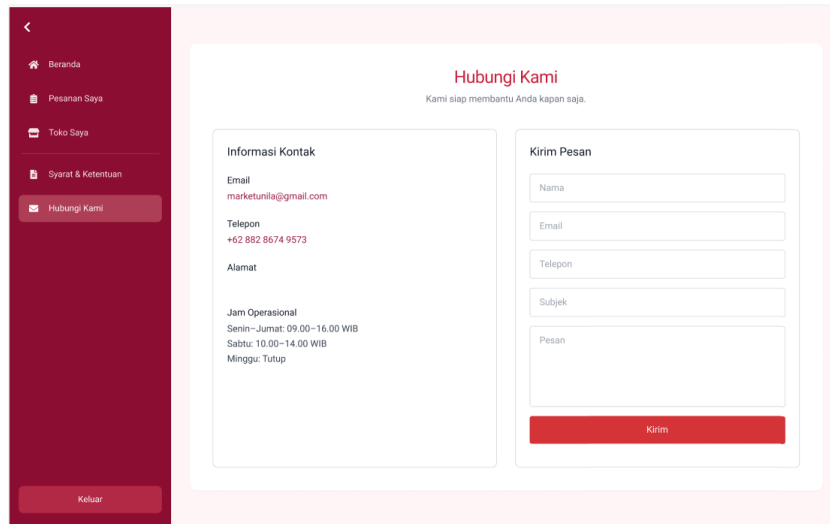
Gambar 3.31 *Mockup* Halaman Syarat & Ketentuan

Gambar 3.31 menampilkan halaman syarat dan ketentuan yang berfungsi untuk menampilkan informasi aturan dan kebijakan penggunaan sistem U-Market. Halaman ini bertujuan memberikan kejelasan hak dan kewajiban pengguna dalam menggunakan layanan yang tersedia pada sistem.



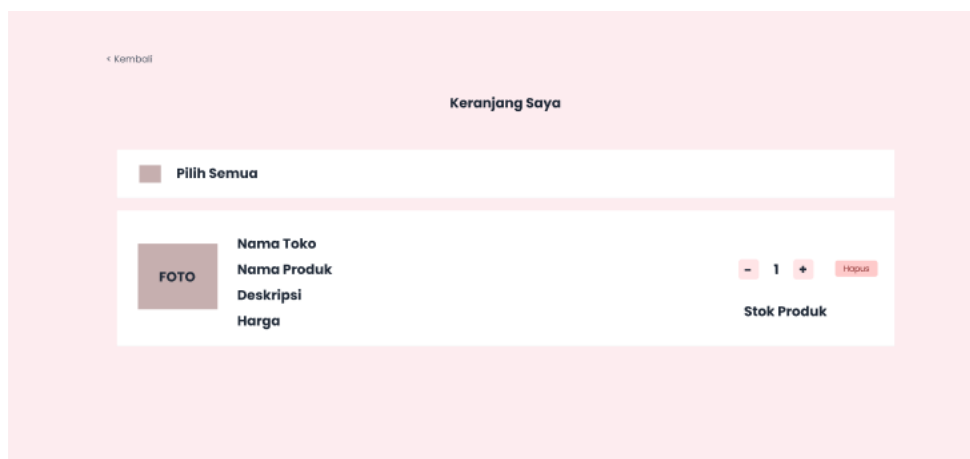
Gambar 3.32 *Mockup* Halaman Edit Profil Pengguna

Gambar 3.32 menampilkan halaman edit profil pengguna yang digunakan untuk mengelola dan memperbarui data akun pengguna. Halaman ini menampilkan informasi profil seperti nama, email, dan kata sandi yang dapat diubah untuk menjaga keakuratan dan keamanan data pengguna.



Gambar 3.33 *Mockup* Halaman Hubungi Kami

Gambar 3.33 menampilkan halaman hubungi kami yang berfungsi sebagai media komunikasi antara pengguna dan pengelola sistem. Halaman ini menampilkan informasi kontak serta form pengiriman pesan yang dapat digunakan pengguna untuk menyampaikan pertanyaan, saran, atau kendala terkait penggunaan sistem.



Gambar 3.34 *Mockup* Halaman Keranjang Saya

Gambar 3.34 menampilkan halaman keranjang saya yang berfungsi untuk menampilkan daftar produk yang telah dipilih oleh pengguna sebelum proses pembayaran. Halaman ini menampilkan informasi produk, jumlah, stok, serta menyediakan aksi pengelolaan *item* guna mendukung proses transaksi.

### 3.3.3 *Sprint Planning*

*Sprint Planning* dilaksanakan pada awal setiap *Sprint* sebagai tahap perencanaan pengembangan sistem. Pada tahap ini, tim pengembang bersama pihak terkait melakukan pembahasan untuk menentukan ruang lingkup pekerjaan yang akan dikerjakan selama *Sprint* berlangsung. Proses ini bertujuan untuk memilih dan menyusun fitur atau pekerjaan dari *Product Backlog* yang akan dikembangkan sesuai dengan prioritas dan kebutuhan sistem.

Dalam pelaksanaan *Sprint Planning*, peran *Scrum Master* adalah memastikan proses perencanaan berjalan dengan baik serta memastikan seluruh tim memahami tujuan *Sprint* yang telah ditetapkan. Hasil dari *Sprint Planning* berupa *Sprint Backlog* yang berisi daftar pekerjaan teknis *back-end* yang akan dikembangkan untuk mencapai tujuan *Sprint* pada sistem *e-commerce* U-Market.

### 3.3.4 *Daily Scrum*

*Daily Scrum* merupakan kegiatan koordinasi yang dilaksanakan secara rutin oleh tim pengembang selama *Sprint* berlangsung. Kegiatan ini difasilitasi oleh *Scrum Master* dan diikuti oleh seluruh anggota tim pengembang dengan durasi singkat, umumnya sekitar 15 menit. Dalam pelaksanaannya, frekuensi dan waktu *Daily Scrum* dapat disesuaikan dengan kebutuhan pengembangan sistem. Tujuan utama dari kegiatan ini adalah untuk meningkatkan kolaborasi tim serta memantau progres pekerjaan dengan mengevaluasi aktivitas yang telah dilakukan sebelumnya dan merencanakan pekerjaan yang akan dikerjakan pada hari berikutnya. Pembahasan dalam *Daily Scrum* berfokus pada capaian pekerjaan sebelumnya, rencana

pekerjaan selanjutnya, serta kendala atau hambatan yang dihadapi selama proses pengembangan.

Pada penelitian ini, proses pengujian sistem dilakukan secara bertahap pada setiap iterasi *Sprint*. Pengujian difokuskan pada pengujian fungsional *back-end* untuk memastikan seluruh layanan REST API dapat berjalan sesuai dengan kebutuhan sistem. Metode pengujian yang digunakan adalah *black box testing*, yaitu metode pengujian yang berfokus pada kesesuaian *output* terhadap *input* tanpa memperhatikan struktur internal kode program.

Pengujian REST API dilakukan menggunakan alat bantu Postman dengan skenario pengujian yang telah ditentukan. Pengujian ini mencakup validasi seluruh *endpoint* yang dikembangkan, seperti memastikan permintaan HTTP yang valid menghasilkan respons yang sesuai, memverifikasi penanganan kesalahan pada permintaan yang tidak valid, serta memastikan sistem dapat memberikan respons yang benar dan konsisten sesuai dengan fungsi yang dirancang. Dengan pengujian ini, diharapkan sistem *back-end e-commerce* U-Market dapat beroperasi secara stabil dan memenuhi kebutuhan fungsional yang telah ditetapkan.

### **3.3.5 *Sprint Review***

*Sprint Review* merupakan tahapan evaluasi yang dilaksanakan pada akhir setiap *Sprint* untuk menilai hasil pengembangan sistem yang telah dicapai selama periode *Sprint* berlangsung. Pada tahap ini, tim pengembang melakukan peninjauan terhadap fitur dan layanan *back-end* yang telah dikembangkan, khususnya pada sistem *e-commerce* U-Market, untuk memastikan kesesuaian antara hasil implementasi dengan tujuan *Sprint* yang telah ditetapkan sebelumnya.

Dalam pelaksanaan *Sprint Review*, *Scrum Team* berinteraksi dengan *reviewer* atau pemangku kepentingan untuk mempresentasikan hasil kerja yang telah dicapai. Kegiatan ini dilakukan melalui demonstrasi fungsional sistem, sehingga *reviewer* dapat memberikan masukan dan umpan balik secara langsung. Masukan yang

diperoleh dari *Sprint Review* selanjutnya digunakan sebagai dasar evaluasi dan perbaikan pada *Sprint* berikutnya guna meningkatkan kualitas dan kesesuaian sistem dengan kebutuhan pengguna.

### **3.3.6 *Sprint Retrospective***

*Sprint Retrospective* merupakan tahapan refleksi yang dilakukan oleh tim pengembang setelah *Sprint* berakhir. Pada tahap ini, tim melakukan evaluasi terhadap proses pengembangan yang telah dijalankan untuk mengidentifikasi hal-hal yang telah berjalan dengan baik maupun kendala yang dihadapi selama *Sprint*. Evaluasi ini bertujuan untuk merumuskan langkah perbaikan yang akan diterapkan pada *Sprint* berikutnya.

Melalui *Sprint Retrospective*, setiap anggota tim memperoleh umpan balik secara cepat terkait kinerja dan proses kerja yang dilakukan. Hasil evaluasi tersebut digunakan sebagai dasar untuk meningkatkan kualitas pengembangan sistem serta memperbaiki pola kerja tim, sehingga pengembangan *back-end e-commerce U-Market* dapat berlangsung lebih efektif dan berkelanjutan pada *Sprint* selanjutnya.

### **3.3.7 Penulisan Laporan**

Setelah seluruh proses penelitian dan pengembangan perangkat lunak selesai dilaksanakan, tahapan selanjutnya adalah penyusunan laporan penelitian. Laporan ini disusun sebagai bentuk dokumentasi hasil penelitian yang telah dilakukan, sekaligus menjadi sumber referensi dan bahan literatur yang dapat digunakan dalam penelitian selanjutnya maupun sebagai rujukan dalam pengembangan sistem yang berkaitan.

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa:

1. Penelitian ini menghasilkan sistem *back-end website e-commerce* U-Market menggunakan *framework* Laravel yang dapat digunakan oleh masyarakat untuk melakukan aktivitas jual beli secara efisien dan aman. Hasil pengujian performa dengan metode *Stress Test* menunjukkan kinerja yang stabil hingga 60 pengguna virtual dengan tingkat *error* 0%, serta masih dapat berjalan pada 150 pengguna virtual dengan *error* sebesar 14,83%. Sehingga sistem yang dibangun telah mampu menangani beban pengguna secara simultan dalam batas tertentu dan dikategorikan layak digunakan.
2. *Framework* Laravel berhasil diterapkan untuk membangun sistem *back-end* dengan arsitektur yang terorganisasi dan modular melalui pemisahan *model*, *controller*, *middleware*, dan *route*. Penerapan Eloquent ORM mempermudah pengelolaan relasi antar tabel, sementara *middleware* dan mekanisme RBAC mendukung pembatasan akses berbasis peran antara admin dan pengguna.
3. Metode *Scrum* berhasil diimplementasikan dalam proses pengembangan sistem melalui pembagian pekerjaan ke dalam empat *Sprint* yang terstruktur. Setiap *Sprint* menghasilkan *increment* fungsional yang dapat diuji dan dievaluasi melalui *Sprint Review* sebelum melanjutkan ke tahap berikutnya, sehingga pengembangan dilakukan secara iteratif dan terukur.
4. Sistem yang dibangun mampu mendukung aktivitas jual beli secara lebih efisien dan terstruktur dibandingkan metode sebelumnya yang menggunakan grup WhatsApp. Proses transaksi, pencatatan pembayaran, serta riwayat pesanan dapat dilakukan secara otomatis dan terdokumentasi dengan baik dalam basis data.

## 5.2 Saran

Berdasarkan hasil penelitian dan pengembangan sistem yang telah dilakukan, maka saran yang dapat diberikan untuk pengembangan selanjutnya adalah sebagai berikut.

1. Melakukan pengembangan sistem dengan membangun aplikasi *mobile* (Android/iOS) yang terintegrasi dengan REST API U-Market. Pengembangan aplikasi *mobile* dapat meningkatkan kenyamanan pengguna dalam melakukan transaksi, memperluas jangkauan pasar, serta memberikan pengalaman pengguna yang lebih optimal dibandingkan akses melalui web *browser*.
2. Menambahkan fitur pelacakan lokasi pesanan (*order tracking*) secara *real-time* untuk memberikan informasi terkait posisi dan status pengiriman barang kepada pengguna. Hal ini dapat meningkatkan transparansi serta memudahkan pemantauan proses pengiriman oleh pembeli maupun penjual.
3. Mengembangkan sistem manajemen kategori dan pencarian produk yang lebih kompleks, seperti filter berdasarkan harga, *rating*, atau lokasi, guna meningkatkan kemudahan pengguna dalam menemukan produk yang diinginkan.

## DAFTAR PUSTAKA

- [1] M. R. Fahlevi dan S. Mu'minin, "Web-Based Online Learning Platform with RAD and laravel Methods," *Ultimatics : Jurnal Teknik Informatika*, vol. 17, no. 1, hlm. 107, Jul 2025.
- [2] D. Permana dan K. Santosa, "Perancangan Sistem Informasi Administrasi Transaksi E-Commerce Menggunakan Metode RAD dan MVC Framework Laravel Pada KOMA Collection," *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 3, no. 5, hlm. 1303–1313, Mei 2024, [Daring]. Tersedia pada: <https://journal.mediapublikasi.id/index.php/oktal>
- [3] A. Sunardi dan Suharjito, "MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based," *Procedia Computer Science*, Elsevier B.V., 2019, hlm. 134–141. doi: 10.1016/j.procs.2019.08.150.
- [4] Y. Candra Kurniawan dan T. Raharjo, "Scrum Effectiveness Measurement in Human Resources Technology Product at Telecommunication Company," *Journal of Informatics and Communications Technology*, vol. 5, no. 1, hlm. 149–158, 2023, doi: 10.52661.
- [5] S. A. Bhat, K. Kansana, dan J. M. Khan, "A Review Paper On E-Commerce," *Asian Journal of Technology & Management Research*, vol. 6, no. 1, hlm. 2249–0892, Feb 2021.
- [6] S. Gupta, P. S. Kushwaha, U. Badhera, P. Chatterjee, dan E. D. R. S. Gonzalez, "Identification of benefits, challenges, and pathways in E-commerce industries: An integrated two-phase decision-making model," *Sustainable Operations and Computers*, vol. 4, hlm. 200–218, Jan 2023, doi: 10.1016/j.susoc.2023.08.005.
- [7] B. Chessa Sarmanela, A. N. Samiya, dan M. E. Yuliana, "Mengenal Backend Developer: Skill, dan Tanggung Jawab Menjadi Backend Developer," *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, 2023.

- [8] B. N. Ramadhan dan S. Sugiyanto, “Analisis *Usability Website* Sistem Informasi Sd Negeri 1 Wangon Menggunakan Metode *System Usability Scale*,” *Jurnal Informatika Teknologi dan Sains*, vol. 6, no. 3, hlm. 421–431, Agu 2024.
- [9] A. C. Tajimalela, R. P. Suryahadi, R. Alfadilla, I. Syawanodya, dan I. Tawakal, “*Comparison Towards Different Methods of Software Debugging*,” *Journal of Software Engineering, Information and Communication Technology (SEICT)*, vol. 5, no. 2, hlm. 113–122, Des 2024, doi: 10.17509/seict.v5i2.70590.
- [10] S. W. Ramdany, S. Aulia Kaidar, B. Aguchino, C. Amelia, A. Putri, dan R. Anggie, “Penerapan UML *Class diagram* dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web,” *Journal of Industrial and Engineering System*, vol. 5, no. 1, hlm. 30–41, Jun 2024.
- [11] F. T. Sheldon, K. Jerath, O. Pilskalns, Y. J. Kwon, W. H. Kim, dan H. Chung, “*Case Study: B2B E-Commerce System Specification and Implementation Employing Use-Case Diagrams, Digital Signatures and XML*,” *IEEE Computer Society*. 2002. [Daring]. Tersedia pada: <http://www.uml.co.kr/>
- [12] N. Dahri, H. Setya Hadi, dan R. Formis, “Jurnal Manajemen Teknologi Informatika Perancangan Sistem *E-Commerce* Berbasis Kemitraan Dagang,” *Jurnal Manajemen Teknologi Informatika*, vol. 1, no. 3, hlm. 176, Des 2023.
- [13] T. M. Connolly dan C. E. Begg, *Database systems : a practical approach to design, implementation and management*. Pearson Education Limited, 2015.
- [14] A. C. Dea, G. Ginting, dan N. P. I. Muhammad, “Analisis Kinerja *Database* Dalam Sistem *E-Commerce*,” *Masip: Jurnal Manajemen Administrasi Bisnis dan Publik Terapan*, vol. 1, no. 3, hlm. 42–45, Jul 2023, doi: 10.59061/masip.v1i3.303.
- [15] F. Luthfi, “Penggunaan *Framework* Laravel Dalam Rancang Bangun Modul *Back-End* Artikel *Website* Bisnisbisnis.ID,” *JISKa*, 2017.
- [16] I. Nurjaman, F. Setyo Utomo, dan N. Hermanto, “Penerapan REST API Laravel sebagai Fondasi *Back-end* Aplikasi G-MOOC 4D”, *Journal of Informatics and Interactive Technology*. Vol.1, no.1, 2024 doi: 10.33395/sinkron.xxx.xxx.
- [17] Michele E. Davis and Jon A. Phillips, *Learning PHP and MySQL. Second Edition*. 2007.

- [18] E. Pangestu Wonohardjo, R. Febriyanto Sunaryo, Y. Sudiyono, dan N. Surantha, "A Systematic Review Of SCRUM In Software Development," *International Journal On Informatics Visualization*, vol. 3, 2019.
- [19] Ian. Sommerville, *Software engineering*. Pearson, 2016.
- [20] P. P. Kore, M. J. Lohar, M. T. Surve, dan S. Jadhav, "API Testing Using Postman Tool," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, hlm. 841–843, Des 2022, doi: 10.22214/ijraset.2022.48030.
- [21] Luh N, Ginasari A, Wibawa K, Kadek N, Wirdiani A, "Penguujian Stress Testing API Sistem Pelayanan dengan Apache JMeter," *Jurnal Ilmiah Teknologi dan Komputer (JITTER)*, Vol. 2, no. 3, 2021.
- [22] K. Zay Oo, "Design and Implementation of Electronic Payment gateway for Secure Online Payment System the Creative Commons Attribution License (CC BY 4.0)," *International Journal of Trend in Scientific Research and Development (IJTSRD) International Journal of Trend in Scientific Research and Development*, vol. 3, no. 5, hlm. 1329–1334, 2019, doi: 10.31142/ijtsrd26635.
- [23] Y. Fatman, N. K. Nafisah, dan P. J. B. Pambudi, "Implementasi Payment Gateway dengan Menggunakan Midtrans pada Website UMKM Geberco," *Jurnal KomtekInfo*, vol. 10, hlm. 64–72, Jun 2023, doi: 10.35134/komtekinfo.v10i2.364.
- [24] F. Ahmad Fauzi dan F. Darmawan, "Pembangunan Aplikasi E-Commerce berbasis Website Menggunakan Laravel," *Pasinformatik*, vol. 2, no. 1, 2023, [Daring]. Tersedia pada: <https://journal.unpas.ac.id/index.php/pasinformatik>
- [25] S. H. Hasibuan, M. Irwan, P. Nasution, S. Suci, dan A. Sundari, "Development Of Payment gateway Digitalization Using Midtrans In The Use Of Halodoc," *International Journal On Advanced Technology, Engineering, And Information System (IJATEIS)*, vol. 2, no. 1, 2023, [Daring]. Tersedia pada: <http://ojs.transpublika.com/index.php/IJATEIS>
- [26] A. Andipradana dan D. K. Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, vol. 19, hlm. 161–172, Agu 2021.
- [27] R. Syahputra, A. Rahman Winardi, A. Rahmadani, R. Islamiah, dan M. L. Hamzah, "Rancang Bangun Sistem Informasi Penjualan Mainan Edukasi Bricksgenius Berbasis Web Menggunakan Metode Scrum," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 2, no. 2, hlm. 98–110, 2024.

- [28] M. Rahmouni, M. Bouzaidi, dan S. Mbarki, “*Approach by modeling to generate an e-commerce web code from laravel model,*” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, hlm. 257–266, Apr 2023, doi: 10.11591/ijeecs.v30.i1.pp257-266.
- [29] M. Maulana, Z. Aditya, W. Hayuhardhika, N. Putra, dan I. Arwani, “Pengembangan Sistem Informasi *E-Commerce* dengan Pemanfaatan API Midtrans menggunakan *Framework* Laravel (Studi Kasus: Byboot.id),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 8, hlm. 3899–3906, Agu 2022, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [30] C. Legisna, A. Ayurira, dan I. N. Fajri, “Implementasi Metode *Scrum* Dalam Pengembangan *Website E-Commerce* Pada Twins Petshop,” *Jurnal Innovation and Future Technology (IFTECH)*, vol. 6, no. 2, hlm. 2656–2774, 2024.
- [31] G. B. Santoso, T. M. Sinaga, dan A. Zuhdi, “*MVC Implementation In Laravel Framework For Development Web-Based E-Commerce Applications,*” *Intelmatics*, vol. 1, no. 1, hlm. 37–42, Jan 2021, doi: 10.25105/itm.v1i1.7867.