

TUGAS AKHIR

RANCANG BANGUN SISTEM INFORMASI PEMASARAN DAN PENJUALAN PRODUK UMKM SELERA INDONESIA BERBASIS *WEB*

Diajukan Sebagai Persyaratan Untuk Menyelesaikan Program Pendidikan

Diploma III Pada Program Studi Teknik Informatika

Jurusan Teknik Elektro Politeknik Negeri Pontianak



DISUSUN OLEH :

MUHAMMAD RIZKY IKHSAN PRATAMA

3202016043

PROGRAM STUDI D-III TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI PONTIANAK

TAHUN 2023

HALAMAN PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI PEMASARAN DAN PENJUALAN PRODUK UMKM SELERA INDONESIA BERBASIS *WEB*

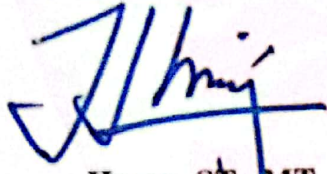
Oleh :

Muhammad Rizky Ikhsan Pratama (3202016043)

Tugas akhir ini telah diterima dan disahkan sebagai salah satu syarat untuk menyelesaikan program Pendidikan Diploma III pada program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Pontianak.

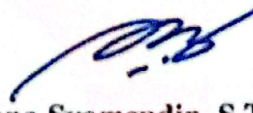
Disahkan oleh :

Ketua Jurusan Teknik Elektro



Hasan, S.T., MT
NIP. 19710820 199903 1 003

Koordinator Program Studi
D-III Teknik Informatika



Mariana Syamsudin, S.T., M.T., PhD
NIP. 197503142006042001

Mengetahui :

Direktur Politeknik Negeri Pontianak



Dr. H. Widodo PS, S.T., M.T.
NIP. 197504242000031001

HALAMAN PERNYATAAN

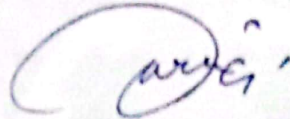
RANCANG BANGUN SISTEM INFORMASI PEMASARAN DAN PENJUALAN PRODUK UMKM SELERA INDONESIA BERBASIS *WEB*

Oleh :

Muhammad Rizky Ikhsan Pratama

3202016043

Dosen Pembimbing :



Fitri Wibowo, S.S.T., M.T.

NIP. 198512282015041002

**Telah dipertahankan di depan penguji pada tanggal 29 Agustus 2023 dan
dinyatakan memenuhi syarat sebagai Laporan Tugas Akhir**

Dosen Penguji :

Penguji 1



Budianingsih, S.T., M.T.
NIP. 198011022012122003

Penguji 2



Sarah Bibi, S.S.T., M.Pd.
NIP. 198806042019092001

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Rizky Ikhsan Pratama

NIM : 3202016043

Jurusan / Program Studi : Teknik Elektro / Teknik Informatika

Judul Proposal : Rancang Bangun Sistem Informasi Pemasaran dan
Penjualan Produk UMKM Selera Indonesia
Berbasis *Web*


Menyatakan dengan sebenar-benarnya bahwa penulisan proposal Tugas Akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah proposal maupun kegiatan yang tercantum sebagai bagian dari proposal Tugas Akhir ini. Jika terdapat karya orang lain, saya akan mencantumkan sumber yang jelas.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di Politeknik Negeri Pontianak.

Demikian pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Pontianak, 29 Agustus 2023

Yang membuat pernyataan,

A handwritten signature in blue ink is written over a red and white revenue stamp. The stamp is a 'Meterai Tempel' (adhesive stamp) for 10,000 Rupiah, with the serial number 96DC6AC048612095. The signature is written in a cursive style.

Muhammad Rizky Ikhsan Pratama

NIM. 3202016043

RIWAYAT HIDUP



Biodata Mahasiswa :

Nama Mahasiswa	: Muhammad Rizky Ikhsan Pratama
NIM	: 3202016043
Tempat/Tanggal Lahir	: Pontianak/13 Februari 2002
Jenis Kelamin	: Laki-laki
Agama	: Islam
Alamat	: Jalan Sungai Raya Dalam Komp. Bhayangkara Permai Blok F No. 11
No.Telpon Rumah/Handphone	: 0895338898951
Email	: rizkyikhsan204@gmail.com

ABSTRAK

Teknologi informasi telah menjadi pendorong utama dalam transformasi bisnis, termasuk pada skala Usaha Mikro Kecil dan Menengah (UMKM), yang semakin mengadopsi penggunaan *website* sebagai sarana informasi. Sebagai contoh, Selera Indonesia, sebuah UMKM yang fokus pada produk khas Kalimantan Barat, memanfaatkan platform digital untuk memperluas pasar secara nasional. Namun, tantangan seperti *branding* yang kurang efektif dan keterbatasan sumber daya manusia menghambat perkembangan Selera Indonesia. Untuk mengatasi hal ini, strategi UMKM dapat meliputi pembangunan *website* sebagai upaya penguatan *branding*, didukung oleh promosi melalui media sosial untuk mengarahkan pengguna ke tautan *website*. Meski demikian, keterbatasan sumber daya manusia tetap menjadi kendala dalam merancang strategi *branding* yang optimal.

Untuk mengatasi tantangan-tantangan ini, pembangunan *website* pemasaran menjadi solusi yang berpotensi memberikan dampak positif bagi Selera Indonesia. Dengan memanfaatkan *framework* Laravel dan MySQL sebagai sistem manajemen *database* dapat memberikan pendekatan yang efisien dalam pengembangan *website*. Proses pengembangan mengikuti metodologi *modified waterfall*. Metodologi *modified waterfall* pendekatan pengembangan perangkat lunak yang menggabungkan elemen model *waterfall* dengan kemungkinan adanya iterasi dan penyesuaian fase untuk fleksibilitas yang lebih besar.

Dengan memanfaatkan teknologi informasi, Selera Indonesia berharap dapat mengatasi tantangan branding yang kurang efektif serta keterbatasan sumber daya manusia yang mempengaruhi perkembangan bisnisnya serta berupaya mengoptimalkan pemanfaatan media sosial sebagai alat promosi yang mengarahkan pengguna ke *website*. Dengan harapan ini, Selera Indonesia berusaha mewujudkan pertumbuhan yang berkelanjutan dan kompetitif dalam lingkungan bisnis yang semakin berkembang.

Kata Kunci : *Website, Branding, Laravel, MySQL*

ABSTRACT

Information technology has become a key driver in business transformation, including at the scale of Micro, Small and Medium Enterprises (UMKM), which are increasingly adopting the use of websites as a means of information. For example, Selera Indonesia, an UMKM focusing on West Kalimantan specialty products, is utilizing digital platforms to expand its market nationwide. However, challenges such as ineffective branding and limited human resources have hindered Selera Indonesia's development. To overcome this, MSME strategies can include website development as an effort to strengthen branding, supported by promotion through social media to direct users to the website link. However, limited human resources remain an obstacle in designing an optimal branding strategy.

To overcome these challenges, the development of a marketing website is a solution that has the potential to have a positive impact on Selera Indonesia. Utilizing the Laravel framework and MySQL as a database management system can provide an efficient approach to website development. The development process follows the modified waterfall methodology. The modified waterfall methodology is a software development approach that combines elements of the waterfall model with the possibility of iteration and phase adjustments for greater flexibility.

By utilizing information technology, Selera Indonesia hopes to overcome the challenges of ineffective branding and limited human resources that affect its business development and seeks to optimize the use of social media as a promotional tool that directs users to the website. With this hope, Selera Indonesia strives to realize sustainable and competitive growth in an increasingly evolving business environment.

Keywords : Website, Branding, Laravel, MySQL

PRAKATA

Puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang telah menganugerahkan banyak nikmat sehingga penulis dapat menyusun laporan Tugas Akhir yang berjudul “Rancang Bangun Sistem Informasi Pemasaran Dan Penjualan Produk UMKM Selera Indonesia Berbasis *Web*” ini dengan baik.

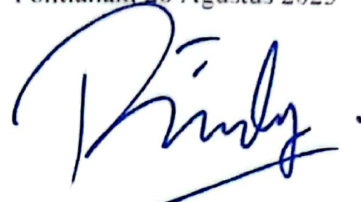
Dalam penyusunan laporan tugas akhir ini, tentu tak lepas dari pengarahan dan bimbingan dari berbagai pihak. Maka dari itu penulis ucapkan rasa hormat dan terima kasih kepada:

1. Kedua orang tua yang telah memberikan nasihat dan dukungan selama proses PKL;
2. Bapak Dr. H. Widodo PS, S.T., M.T. selaku Direktur Politeknik Negeri Pontianak;
3. Bapak Hasan, ST., MT selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Pontianak;
4. Ibu Mariana Syamsudin, S.T., M.T., PhD selaku Koordinator Program Studi D-III Teknik Informatika Politeknik Negeri Pontianak;
5. Bapak Fitri Wibowo, S.S.T., M.T. selaku Koordinator Tugas Akhir Program Studi Teknik Informatika dan sekaligus dosen pembimbing yang telah memberikan arahan, dorongan serta masukan kepada penulis ;
6. Bapak Budianingsih, S.T., M.T. selaku dosen Penguji 1;
7. Bapak Sarah Bibi, S.S.T., M.Pd. selaku dosen Penguji 2;
8. Dosen Program Studi Diploma III Teknik Informatika Politeknik Negeri Pontianak yang telah memberi bekal ilmu kepada penulis.
9. Seluruh staf pengajar dan administrasi Politeknik Negeri Pontianak, khususnya di Jurusan Teknik Elektro Program Studi Teknik Informatika.
10. Teman-teman Program Studi Teknik Informatika yang selalu memberi semangat, bantuan, dan saran selama proses penyusunan Tugas Akhir ini serta lain-lainya yang tidak dapat disebutkan satu-persatu.

Penulis menyadari masih banyak kekurangan yang terdapat pada laporan ini dengan keterbatasan pengetahuan dan kemampuan dari penulis. Oleh karena itu,

penulis sangat mengharapkan kritik dan saran dari pembaca yang membangun untuk menjadi bahan pelajaran di kemudian hari. Semoga laporan Tugas Akhir (TA) ini dapat memberikan manfaat bagi kita semua.

Pontianak, 28 Agustus 2023



Muhammad Rizky Ikhsan Pratama

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN	iii
RIWAYAT HIDUP.....	v
ABSTRAK.....	vi
ABSTRACT.....	vii
PRAKATA.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN	15
1.1. Latar Belakang	15
1.2. Rumusan Masalah.....	17
1.3. Batasan Masalah	17
1.4. Tujuan Penelitian	18
1.5. Manfaat Penelitian	18
1.5.1. Bagi UMKM Selera Indonesia.....	18
1.5.2. Bagi Masyarakat.....	19
1.5.3. Bagi Penulis	19
1.6. Metodologi Penelitian.....	19
1.7. Sistematika Penulisan	24
BAB II LANDASAN TEORI	25
2.1. Tinjauan Pustaka.....	25
2.2. Dasar Teori.....	26
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	27
3.1. Menentukan Project Requirements	27
3.1.1 Use Case Diagram	28
3.1.2 Desain Wireframe.....	29
3.1.3 Merancang Diagram Database	37
BAB IV HASIL DAN PEMBAHASAN	39
4.1. Implementasi Antarmuka Pengguna	39

4.1.1	Implementasi Desain ke Dalam Proyek	39
4.1.2	Pengujian.....	84
4.2.	Implementasi Akhir (<i>Cutover</i>).....	91
4.3.	Hasil Pengujian Seluruh Sistem.....	91
BAB V PENUTUP.....		92
5.1.	Kesimpulan	92
5.2.	Saran	93
DAFTAR PUSTAKA		94

DAFTAR GAMBAR

Gambar 1.1 Model <i>Modified Waterfall</i>	20
Gambar 3.1 <i>Use Case Diagram</i>	28
Gambar 3.2 <i>Wireframe</i> Halaman Beranda	29
Gambar 3.3 <i>Wireframe</i> Halaman Produk.....	30
Gambar 3.4 <i>Wireframe</i> Halaman Akun <i>Login</i>	31
Gambar 3.5 <i>Wireframe</i> Halaman Daftar Akun	31
Gambar 3.6 <i>Wireframe</i> Halaman Beranda Akun <i>Login</i>	32
Gambar 3.7 <i>Wireframe</i> Halaman Transaksi.....	33
Gambar 3.8 <i>Wireframe</i> Halaman <i>Dashboard</i>	34
Gambar 3.9 <i>Wireframe</i> Halaman Produk.....	35
Gambar 3.10 <i>Wireframe</i> Halaman Kategori	36
Gambar 3.11 <i>Wireframe</i> Halaman Transaksi.....	37
Gambar 3.12 <i>Diagram Database</i>	38
Gambar 4.1 Tampilan Halaman Beranda.....	41
Gambar 4.2 Tampilan Halaman Produk	43
Gambar 4.3 Tampilan Halaman <i>Login</i>	45
Gambar 4.4 Tampilan Halaman Daftar Akun	47
Gambar 4.5 Tampilan Halaman Beranda Akun <i>Login</i>	50
Gambar 4.6 Tampilan Rincian Produk	51
Gambar 4.7 Tampilan Halaman Transaksi	64
Gambar 4.8 Menampilkan Total Pembayaran di Midtrans.....	66
Gambar 4.9 Tampilan Ketika Transaksi Berhasil	66
Gambar 4.10 Tampilan Halaman <i>Dashboard</i> Admin	71
Gambar 4.11 Tampilan Halaman Produk	72
Gambar 4.12 Tampilan Halaman Kategori	75
Gambar 4.13 Tampilan Halaman Transaksi	78
Gambar 4.14 Tampilan Halaman Akun Pengguna	81

Gambar 4.15 Pengujian SUS 1	86
Gambar 4.16 Pengujian SUS 2	87
Gambar 4.17 Pengujian SUS 3	88
Gambar 4.18 Pengujian Oleh <i>Stakeholder</i>	90
Gambar 4.19 Pengujian Pada Bagian Midtrans dan Raja Ongkir	90

DAFTAR TABEL

Tabel 4.1 Pertanyaan SUS	85
Tabel 4.2 Hasil SUS.....	89
Tabel 4.3 <i>SUS Acceptability Score</i>	89

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan pesat dalam teknologi *modern* saat ini telah memungkinkan inovasi dalam berbagai bidang, seperti kuliner, pariwisata, pendidikan, politik, perdagangan, perkantoran, dan lainnya, dengan tujuan untuk meningkatkan efisiensi dalam pekerjaan yang sebelumnya memerlukan usaha yang besar [1]. Teknologi informasi, sebagai salah satu contoh, telah diterapkan secara luas di berbagai sektor. Salah satunya adalah penggunaan *website* oleh UMKM yang merupakan bagian perekonomian nasional untuk meningkatkan kesejahteraan, seperti Selera Indonesia, untuk memberikan akses informasi yang mudah kepada masyarakat [2].

Selera Indonesia adalah contoh UMKM di Kalimantan Barat yang menjual produk lokal, termasuk batik, handicraft, dan aksesoris, dengan potensi untuk memperluas pangsa pasar secara nasional melalui *platform online*. Namun, Selera Indonesia menghadapi beberapa kendala yang menghambat pertumbuhannya. Salah satunya adalah masalah *branding* yang disebabkan oleh penggunaan media sosial sederhana seperti WhatsApp, Facebook, dan Instagram. Solusi yang bisa dipertimbangkan adalah pembuatan *website* sendiri untuk memperkuat *branding* dan meningkatkan visibilitas produk.

Tantangan lain yang dihadapi Selera Indonesia adalah keterbatasan sumber daya manusia dalam hal *branding*. Karena banyak UMKM di Indonesia belum memiliki personil yang terlatih dalam branding, maka diperlukan usaha lebih besar untuk mengatasi masalah ini dan membuat strategi branding yang efektif. Selain itu, biaya untuk merekrut karyawan yang memiliki kemampuan dalam hal *branding* juga bisa menjadi beban bagi UMKM yang memiliki sumber daya yang terbatas.

Untuk mengatasi kendala-kendala tersebut, Selera Indonesia membutuhkan *website* pemasaran yang dapat membantu usaha ini untuk meningkatkan visibilitas dan jangkauan pasar, menyediakan informasi dan layanan yang lengkap dan mudah diakses, serta memanfaatkan teknologi digital untuk mengoptimalkan usaha.

Dengan *website* pemasaran, Selera Indonesia berharap dapat meningkatkan omset, loyalitas pelanggan, kualitas produk, produktivitas, inovasi, dan daya saing usaha.

Dikutip dari hasil penelitian Sudi Anang Triana yang berjudul “Perancangan *Website Sweda Customring* Sebagai Upaya Membangun *Brand Image*” [3], penerapan *website* pada suatu unit usaha mampu memberikan tampilan secara luas dan memberikan fitur yang lebih banyak cangkupannya daripada media sosial Instagram.

Dalam konteks ini, pembangunan *website* sistem informasi pemasaran produk UMKM Selera Indonesia dengan menggunakan *framework* Laravel dan MySQL dapat membantu UMKM dalam mempromosikan produk dan meningkatkan daya saing. Menurut skripsi yang berjudul “Rancang Bangun Sistem Penjaringan Umpan Balik Tingkat Kepuasan Pemangku Kepentingan Menggunakan Arsitektur *Model View Controller*” yang dilakukan oleh Muhammad Shadiq dari Universitas Hasanuddin [4], menjelaskan bahwa Laravel dan MySQL adalah dua teknologi yang sering digunakan untuk mengembangkan aplikasi *website*. Laravel adalah salah satu *framework* PHP yang paling populer dan memiliki banyak fitur dan *library* yang memudahkan proses pengembangan aplikasi *website*. Sedangkan MySQL adalah salah satu *Database Management System* (DBMS) yang sering digunakan bersama dengan PHP. Penggunaan kedua teknologi ini dapat memudahkan proses pembuatan *website* sistem informasi produk UMKM Selera Indonesia.

Website Selera Indonesia tidak hanya menyediakan informasi tentang produk-produk khas Kalimantan Barat, tetapi juga memungkinkan pelanggan untuk melakukan transaksi jual beli secara *online* maka terdapat fitur transaksi. Dengan fitur transaksi ini, pelanggan dapat memilih produk yang diinginkan, menambahkan ke keranjang belanja, mengisi data pengiriman, dan memilih metode pembayaran. Fitur transaksi ini juga memberikan kemudahan bagi usaha Selera Indonesia untuk mengelola stok produk, laporan penjualan, dan akuntansi usaha. Dengan demikian, fitur transaksi ini dapat meningkatkan efisiensi dan kenyamanan bagi pelanggan dan usaha Selera Indonesia.

Selera Indonesia dapat membandingkan dirinya dengan *web* UMKM buatan tugas akhir mahasiswa yang memiliki fitur transaksi. Salah satu contoh *website*

UMKM yang dapat dijadikan referensi adalah *website* e-commerce UMKM Nena Namo [5]. *Website* ini memiliki fitur transaksi yang memungkinkan pelanggan untuk melakukan pembelian produk secara *online*, mengedit halaman *website* sesuai keinginan, dan mendapatkan informasi transaksi melalui SMS. *Website* Selera Indonesia dapat belajar dari *website* tersebut untuk menambahkan fitur transaksi yang dapat meningkatkan penjualan dan loyalitas pelanggan.

Pembangunan *website* sistem informasi produk UMKM Selera Indonesia dengan menggunakan *framework* Laravel dan MySQL serta dilengkapi dengan fitur transaksi merupakan salah satu contoh aplikasi *website* yang mengimplementasikan teknologi terkini. Hal ini diharapkan dapat membantu UMKM dalam meningkatkan daya saing dan memperluas pasar. Selain itu, penggunaan teknologi terkini juga diharapkan dapat meningkatkan efisiensi dan efektivitas proses bisnis UMKM.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam Tugas Akhir ini adalah bagaimana merancang serta membangun sistem informasi pemasaran dan penjualan Selera Indonesia berbasis *web* dengan menggunakan *framework* yaitu Laravel dan basis data yang digunakan adalah MySQL dengan menggunakan fitur transaksi.

1.3. Batasan Masalah

Agar memudahkan pembahasan dari penelitian, maka diperlukan sebuah batasan masalah yang penulis uraikan antar lain sebagai berikut :

1. Penelitian ini dilakukan pada bulan Maret hingga Agustus 2023 dengan subjek pengujian terbatas pada pengguna internal yaitu pemilik dari Selera Indonesia. Penelitian ini tidak mencakup pengujian secara luas pada masyarakat umum atau pengguna eksternal.
2. Pembuatan *website* ini menggunakan *framework* yaitu Laravel dan menggunakan basis data MySQL.
3. Fitur pembayaran pada sistem informasi pemasaran dan penjualan Selera Indonesia akan diintegrasikan menggunakan Midtrans.

4. Fitur perhitungan ongkos kirim pada sistem informasi pemasaran dan penjualan Selera Indonesia akan diintegrasikan menggunakan RajaOngkir.
5. Fitur pemesanan pada sistem informasi pemasaran dan penjualan Selera Indonesia
6. Penelitian ini hanya menguji fungsi dan tampilan dari fitur transaksi *online*, serta fitur katalog produk dari *website* sistem informasi Selera Indonesia yang dibangun menggunakan *framework* Laravel dan basis data MySQL.

1.4. Tujuan Penelitian

Adapun tujuan dari tugas akhir pemasaran dan penjualan produk Selera Indonesia berbasis *web* dapat diuraikan sebagai berikut :

1. Membuat sistem informasi yang dapat menerima transaksi pembelian produk secara *online* dari pelanggan.
2. Membuat sistem informasi yang memiliki fitur transaksi dan kategori produk dengan menggunakan *framework* Laravel. Membuat *web* yang dapat mengelola data pelanggan, produk, pesanan, dan pembayaran secara efisien dan aman dengan menggunakan basis data MySQL.
3. Menerapkan model pengembangan *modified waterfall* untuk merancang dan membangun sistem informasi Selera Indonesia dengan menggunakan *framework* Laravel dan basis data MySQL.

1.5. Manfaat Penelitian

1.5.1. Bagi UMKM Selera Indonesia

Adapun manfaat yang didapatkan oleh UMKM Selera Indonesia dalam pembuatan sistem informasi ini adalah memperluas jangkauan pemasaran produk UMKM, meningkatkan kepercayaan dan loyalitas pelanggan, dan mendapatkan informasi serta data yang bermanfaat guna pengambilan keputusan bisnis.

1.5.2. Bagi Masyarakat

Adapun manfaat yang didapatkan oleh masyarakat yaitu dengan adanya *website* sistem informasi yang profesional dan mudah diakses, masyarakat akan lebih percaya terhadap UMKM Selera Indonesia dan produk-produknya.

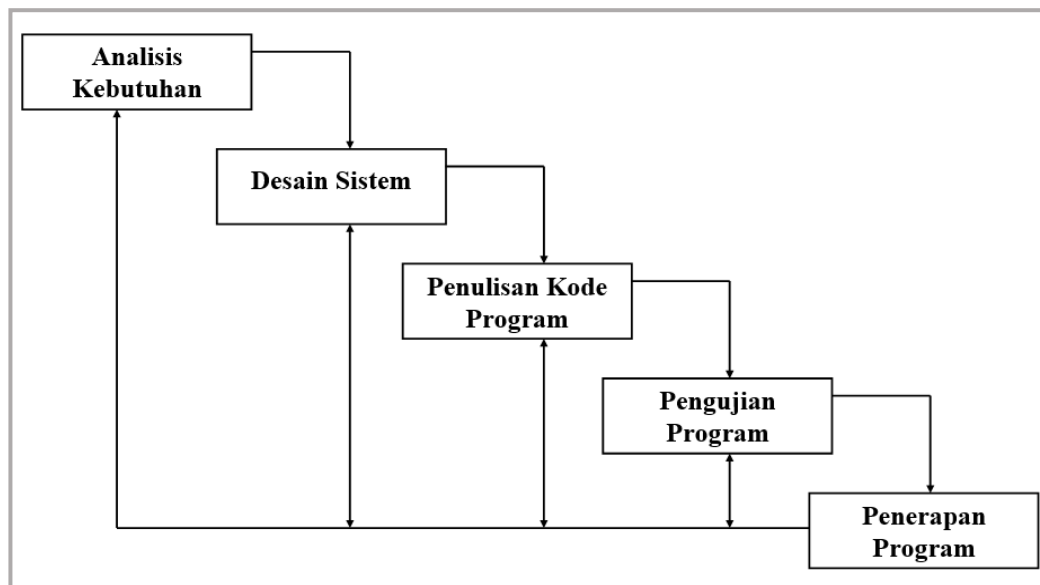
1.5.3. Bagi Penulis

Adapun manfaat yang didapatkan oleh penulis adalah mendapatkan pengalaman dalam merancang, membangun, dan menguji sistem informasi berbasis *web* menggunakan teknologi terkini, serta meningkatkan kemampuan dalam menganalisis dan mengambil keputusan berdasarkan hasil penelitian yang telah dilakukan. Selain itu, penulis juga dapat memperluas wawasan dan pengetahuan mengenai teknologi informasi dan bisnis UMKM.

1.6. Metodologi Penelitian

Adapun model yang akan digunakan dalam pembuatan *website* Selera Indonesia ini adalah Model *Modified Waterfall*. Metodologi ini dipilih karena memiliki tahapan yang terstruktur dan sistematis, sehingga memudahkan pengembangan *website* Selera Indonesia secara bertahap dan dapat terukur. Selain itu, model ini juga memungkinkan dilakukannya evaluasi pada setiap tahapan pengembangan sehingga dapat mengurangi risiko kesalahan dan menjamin kualitas dari sistem

informasi yang dibangun. Maka berikut ini penjelasan dari metodologi model *modified waterfall*.



Gambar 1.1 Model *Modified Waterfall*

Berdasarkan gambar 1.1 Model *Modified Waterfall*, model ini memberikan urutan langkah-langkah pengembangan yang teratur dengan fase iteratif yang fleksibel yang akan memfasilitasi kecukupan dan relevansi dokumentasi dan referensi sehingga memastikan kualitas, keandalan, dan kemudahan pemeliharaan sistem yang dikembangkan [6].

Model ini digambarkan sebagai proses pengembangan perangkat lunak yang bersifat iteratif. Biasanya, dalam model *modified waterfall* lama, setiap fase dalam proses pengembangan dimulai hanya setelah fase sebelumnya selesai, di mana hasilnya digunakan sebagai masukan untuk fase berikutnya secara berurutan. Model ini memungkinkan fase sebelumnya dikembalikan untuk diverifikasi atau divalidasi, yang idealnya terbatas pada langkah-langkah yang terhubung.

Untuk menggunakan *framework* Laravel dan basis data MySQL dengan fitur transaksi, maka perlu mengikuti langkah-langkah berikut:

- Analisis kebutuhan

Pada fase ini, informasi tentang apa yang dibutuhkan dan diharapkan dari sistem Selera Indonesia dikumpulkan dari *stakeholder* dan pengguna akhir, serta batasan dan asumsi yang berlaku untuk proyek ditentukan. Tujuan dari fase ini adalah untuk mengumpulkan informasi tentang

kebutuhan bisnis, teknis, dan fungsional dari sistem yang akan dikembangkan. Pada fase ini terdapat beberapa langkah yang akan dilakukan yaitu mempelajari kebutuhan dan harapan dari *stakeholder* yaitu pemilik Selera Indonesia dan pengguna akhir yaitu konsumen dan *admin website*, mengumpulkan informasi tentang kebutuhan bisnis, teknis, dan fungsional dari sistem yang akan dikembangkan, menentukan batasan dan asumsi yang berlaku untuk tugas akhir. Dari hasil analisa di atas, maka terdapat kebutuhan yaitu:

- *Website* ini dapat menampilkan produk Selera Indonesia
- *Website* ini terdapat fitur pencarian
- *Website* ini terdapat fitur pemesanan
- *Website* ini menyediakan keranjang belanja
- *Website* ini menyediakan sistem pembayaran

- **Desain Sistem**

Fase ini melibatkan pembuatan desain arsitektur sistem yang menunjukkan interaksi dan komunikasi antara komponen sistem, serta pemilihan teknologi yang tepat untuk mengimplementasikan desain arsitektur seperti menggunakan Laravel sebagai *framework* PHP dan MySQL sebagai basis data relasional. Selain itu, desain antarmuka pengguna (*UI*) menggunakan Figma dan desain basis data (*DB*) menggunakan Dbdiagram.io yang sesuai dengan persyaratan fungsional dan non-fungsional juga harus dibuat. Tujuan dari fase ini adalah untuk menghasilkan desain yang memenuhi kebutuhan dan persyaratan yang telah ditetapkan dalam fase analisis kebutuhan. Pada fase ini terdapat beberapa langkah yang akan dilakukan yaitu:

- Membuat desain arsitektur sistem yang menunjukkan interaksi dan komunikasi antara komponen sistem.
- Memilih teknologi yang tepat untuk mengimplementasikan desain arsitektur seperti menggunakan Laravel sebagai *framework* PHP dan MySQL sebagai basis data relasional.
- Membuat desain antarmuka pengguna (*UI*) dan desain basis data (*DB*) yang sesuai dengan persyaratan fungsional dan non-

fungsional.

- **Penulisan Kode Program**

Fase ini melibatkan pengkodean sistem menggunakan Laravel dan MySQL sesuai dengan desain arsitektur, *UI*, dan *DB* yang telah dibuat sebelumnya. Tujuannya adalah memastikan bahwa kode yang dihasilkan bersih, rapi, mudah dibaca, dan sesuai dengan standar pemrograman. Selain itu, fitur transaksi juga harus digunakan untuk menjamin integritas data dalam operasi basis data. Dengan fitur transaksi, sekelompok operasi basis data dapat dilakukan sebagai satu kesatuan kerja yang atomik (tidak dapat dibagi), konsisten (tidak melanggar aturan integritas), terisolasi (tidak dipengaruhi oleh operasi lain), dan tahan lama (tetap ada setelah operasi selesai). Tujuan dari fase ini adalah membangun sistem secara terstruktur dan efektif. Pada fase ini terdapat beberapa langkah yang akan dilakukan yaitu:

- Melakukan pengkodean sistem menggunakan Laravel dan MySQL sesuai dengan desain arsitektur, UI, dan DB yang telah dibuat sebelumnya.
- Memastikan bahwa kode yang dihasilkan bersih, rapi, mudah dibaca, dan sesuai dengan standar pemrograman.
- Menggunakan fitur transaksi untuk menjamin integritas data dalam operasi basis data.

- **Pengujian**

Fase ini ditujukan untuk memeriksa apakah sistem bekerja sesuai dengan persyaratan fungsional dan non-fungsional yang telah ditetapkan. Tujuannya adalah untuk menemukan dan memperbaiki kesalahan atau kecacatan dalam sistem sebelum diterapkan secara luas. Adapun beberapa jenis pengujian yang dapat dilakukan dalam fase ini:

- **Pengujian unit:** Pengujian ini ditujukan untuk memverifikasi fungsi setiap komponen secara terpisah pada tingkat komponen individu.
- **Pengujian integrasi:** Pengujian ini ditujukan untuk memverifikasi bahwa komponen bekerja dengan baik bersama-sama pada tingkat

interaksi antara komponen.

- Pengujian sistem: Pengujian ini ditujukan untuk memverifikasi bahwa semua persyaratan fungsional dan non-fungsional dipenuhi oleh sistem pada tingkat keseluruhan sistem.
- Pengujian penerimaan: Pengujian ini ditujukan untuk mendapatkan umpan balik dari *stakeholder* atau pengguna akhir tentang apakah sistem sesuai dengan kebutuhan atau harapan mereka pada tingkat *stakeholder* atau pengguna akhir.

Pada fase ini terdapat beberapa langkah yang akan dilakukan yaitu:

- Memeriksa apakah sistem bekerja sesuai dengan persyaratan fungsional dan non-fungsional yang telah ditetapkan.
- Melakukan pengujian unit, integrasi, sistem, dan penerimaan.
- Menemukan dan memperbaiki kesalahan atau kecacatan dalam sistem sebelum diterapkan secara luas.

- Penerapan Program

Fase penerapan program akan melibatkan instalasi, konfigurasi, dan pengujian kembali sistem pada infrastruktur yang sesuai untuk memastikan sistem berfungsi dengan baik dalam lingkungan produksi. Selain itu, pelatihan akan diberikan kepada pengguna untuk memastikan penggunaan sistem secara efektif dan efisien. Setelah sistem siap, pengguna akan dapat mengoperasikan sistem. Sistem akan dipantau dan di evaluasi selama operasional untuk memenuhi kebutuhan pengguna dan menjamin operasi yang baik. Jika masalah terdeteksi, sistem akan diperbaiki dan dikembangkan, meskipun pada pelaksanaan tugas akhir fase ini tidak mencakup perbaikan dan pengembangan sistem karena akan diambil alih oleh tim pengembang Selera Indonesia. Fase ini akan berakhir ketika sistem dapat beroperasi secara stabil dan memenuhi kebutuhan pengguna. Pada fase ini terdapat beberapa langkah yang akan dilakukan yaitu:

- Melakukan instalasi, konfigurasi, dan pengujian kembali sistem pada infrastruktur yang sesuai untuk memastikan sistem berfungsi dengan baik dalam lingkungan produksi.
- Memberikan pelatihan kepada pengguna untuk memastikan

penggunaan sistem secara efektif dan efisien.

- Memantau dan mengevaluasi sistem selama operasional untuk memenuhi kebutuhan pengguna dan menjamin operasi yang baik.

1.7. Sistematika Penulisan

Adapun sistem penulisan Tugas Akhir ini secara garis besarnya terbagi menjadi 5 (lima) bab, sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan diuraikan beberapa hal seperti latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

BAB II : LANDASAN TEORI

Pada bab ini akan diuraikan hal-hal yang mengenai Berisi teori-teori dasar relevan dengan masalah yang dibahas dalam Tugas Akhir. Landasan teori juga dapat digunakan sebagai dasar analisis masalah. Untuk proses pengutipan pustaka akan dijelaskan di halaman lain dalam pedoman ini.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Bab ini hanya digunakan untuk mahasiswa yang mengambil Tugas Akhir tentang membuat sistem informasi atau membuat alat, jika Tugas Akhir yang dibuat hanya melakukan analisa terhadap sistem yang sudah ada atau masalah lainnya, judul bab ini dan isinya dapat disesuaikan.

BAB IV : HASIL DAN PEMBAHASAN

Dalam bab ini dilaporkan hasil-hasil yang diperoleh dalam penelitian dan pembahasan terhadap hasil yang telah dicapai maupun masalah-masalah yang ditemui selama penelitian, uji coba, termasuk kelemahan dan kelebihan sistem yang dibuat.

BAB V : PENUTUP

Kesimpulan diambil didasarkan pada tujuan analisis, rincian masalah dan hasil pemecahan masalah. Saran-saran berisi tentang masalah baru yang timbul sehubungan dengan pemecahan masalah.

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penulis mendapatkan beberapa referensi yang dapat dijadikan sebagai pembanding dengan tinjauan Tugas Akhir ini. Penelitian pertama berjudul “Rancang Bangun Sistem Informasi Pemasaran Batik Sembogem Pada UMKM Desa Bogem Magetan” dilakukan oleh Prinda Magdalena dan Andi Rahman Putera dari Universitas PGRI Madiun [7]. Sistem informasi tersebut bertujuan untuk mempermudah pemasaran pada UMKM Batik Sembogem di Desa Bogem berbasis *website*. Sistem informasi ini menggunakan metode *waterfall* dalam pengembangannya dan *framework* Codeigniter dalam implementasinya. Sistem informasi ini memiliki fitur-fitur seperti katalog produk, keranjang belanja, pembayaran *online*, testimoni pelanggan, dan laporan penjualan.

Penelitian kedua berjudul “Sistem Informasi Pemasaran Usaha Kecil Menengah (UKM) Berbasis *Website*” yang dibuat oleh Lilis Indrayani, Feri Awopi, dan Suriani Zabir [8]. Penelitian tersebut membahas tentang pengembangan sistem informasi pemasaran untuk Usaha Kecil Menengah (UKM) di Desa Tritiro Kabupaten Bulukumba yang berbasis *web*. Sistem informasi ini bertujuan untuk mempermudah masyarakat melihat produk-produk yang terdapat pada UKM tertentu dan dapat melakukan pembelian produk untuk wilayah Sulawesi Selatan dan admin dalam mengolah pembelian produk dari pelanggan dan melihat laporan penjualan. Penelitian ini juga menjelaskan tentang proses perancangan sistem informasi pemasaran berbasis *web* dengan menggunakan UML (*Unified Modeling Language*) yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

Dari beberapa referensi yang didapatkan di atas, dapat dijadikan sebagai contoh karena fitur-fitur yang di implementasikan pada referensi-referensi tersebut dapat sesuai dengan fitur yang akan dirancang pada *website* Selera Indonesia tersebut sehingga hal tersebut dapat membantu dalam menghasilkan *website* yang dapat berjalan dengan semestinya.

2.2. Dasar Teori

Berikut ini dasar teori-teori penunjang untuk penulisan Tugas Akhir:

a. Sistem Informasi

Sistem informasi berbasis *web* adalah sistem yang memudahkan pengguna untuk mengakses informasi melalui internet. Sistem ini dapat memberikan informasi dan laporan-laporan yang lebih cepat. Contoh dari sistem informasi berbasis *web* adalah Sistem Informasi Akademik Berbasis *Web* pada Jurusan Sistem Informasi Fakultas Ilmu Komputer di Universitas Sriwijaya. Sistem informasi berbasis *web* juga dapat digunakan dalam kegiatan sekolah, seperti mengelola data calon siswa baru [9].

b. Website

Website adalah kumpulan halaman *web* yang saling berkaitan untuk menyajikan informasi tertentu, yang bisa diakses di internet menggunakan *web* browser [10]; *Website* berisi informasi dalam bentuk teks, gambar, animasi, suara atau gabungan dari semuanya [11].

c. Framework

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi *MIT*, dibangun dengan konsep *Model View Controller (MVC)*. Laravel adalah pengembangan *website* berbasis *MVC* yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan [12].

d. Database Management System

Database Management System (DBMS) adalah sebuah *software* yang dirancang khusus untuk menghubungkan *database* dengan pengguna agar proses pengolahan data dapat dilakukan [13]; *DBMS* menyediakan antarmuka yang memungkinkan pengguna untuk mengakses, memodifikasi, dan mengelola data dalam *database* [14]; *DBMS* juga bertugas untuk memastikan integritas data dan keamanan informasi.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Menentukan Project Requirements

Dalam fase penentuan kebutuhan proyek, langkah-langkah berikut akan dilakukan untuk mengumpulkan informasi tentang kebutuhan bisnis, teknis, dan fungsional dari sistem yang akan dikembangkan. Hal ini mencakup memahami harapan dan kebutuhan dari pemilik Selera Indonesia dan pengguna akhir, seperti konsumen dan admin *website*, serta menetapkan batasan dan asumsi yang berlaku untuk proyek. Hasil dari analisis ini adalah identifikasi kebutuhan berikut:

1. Sistem Selera Indonesia harus memiliki sebuah *website* yang dapat menampilkan produk-produk Indonesia.
2. Website ini harus menyediakan fitur pencarian agar pengguna dapat dengan mudah mencari produk yang mereka inginkan.
3. Diperlukan fitur pemesanan produk melalui *website*.
4. *Website* harus memiliki keranjang belanja agar pengguna dapat mengumpulkan produk yang ingin dibeli sebelum melakukan pembayaran.
5. Sistem juga perlu menyediakan opsi sistem pembayaran yang aman dan efisien.

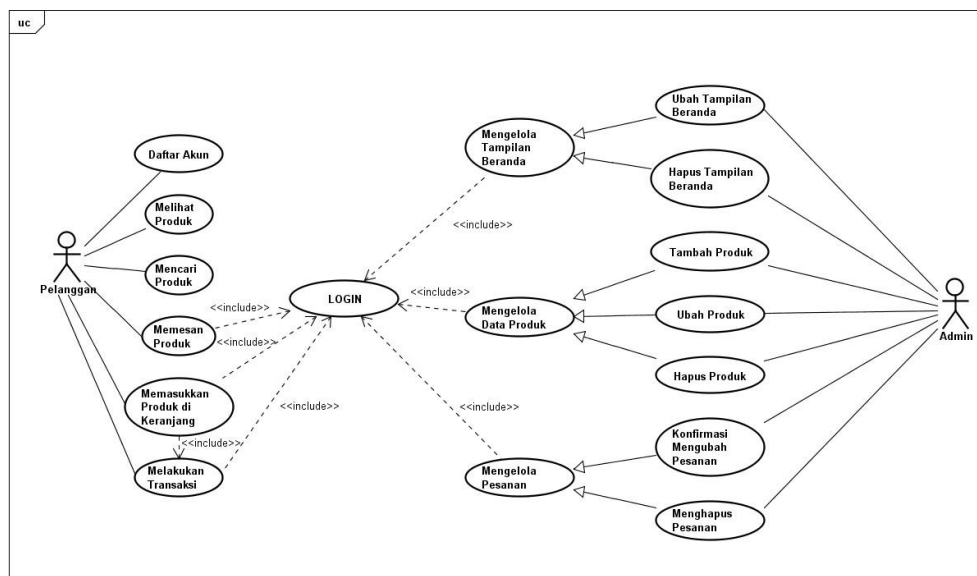
Asal mula penjualan selera Indonesia menjual produk seperti batik, aksesoris, dan lainnya menggunakan teknik *branding share story* sosmed dengan menampilkan produk-produk unggulan dengan penjelasan yang detail serta menargetkan pasar yang lebih luas. Maka penting untuk memperhatikan aspek-aspek desain dan fungsionalitas *website* yang dapat memperkuat *branding* Selera Indonesia dan memudahkan pengguna untuk berinteraksi dengan *website*. Dalam rancangan sistem, perlu dipertimbangkan elemen-elemen seperti tata letak, warna, *font*, dan gambar yang dapat menarik perhatian pengguna dan menciptakan pengalaman pengguna yang nyaman dan mudah digunakan.

Selain itu, perlu juga memperhatikan fitur-fitur yang dibutuhkan pengguna seperti fitur pencarian, keranjang belanja, dan sistem pembayaran yang aman dan mudah digunakan. Dalam rancangan sistem, juga perlu dipertimbangkan integrasi dengan *platform* sosial media dan teknik pemasaran *online* lainnya untuk

memperkuat *branding* dan meningkatkan visibilitas Selera Indonesia di pasar yang lebih luas. Semua aspek ini perlu dipertimbangkan secara matang dalam rancangan sistem berikut ini:

3.1.1 Use Case Diagram

Use case diagram adalah representasi grafis dari *use case* yang digunakan untuk memvisualisasikan hubungan antara aktor (pengguna sistem) dan *use case* (skenario penggunaan). *Use case diagram* memperlihatkan aktor yang terlibat dalam sistem dan skenario penggunaan yang berbeda di mana mereka terlibat. Berdasarkan analisa yang dilakukan dengan mengusulkan sistem informasi pemasaran berbasis *web* yang akan dirancang bertujuan untuk mempermudah pemasaran dan transaksi pada UMKM Selera Indonesia. Maka dirancang desain alur sistem dengan aktor melalui *use case diagram* berikut ini :

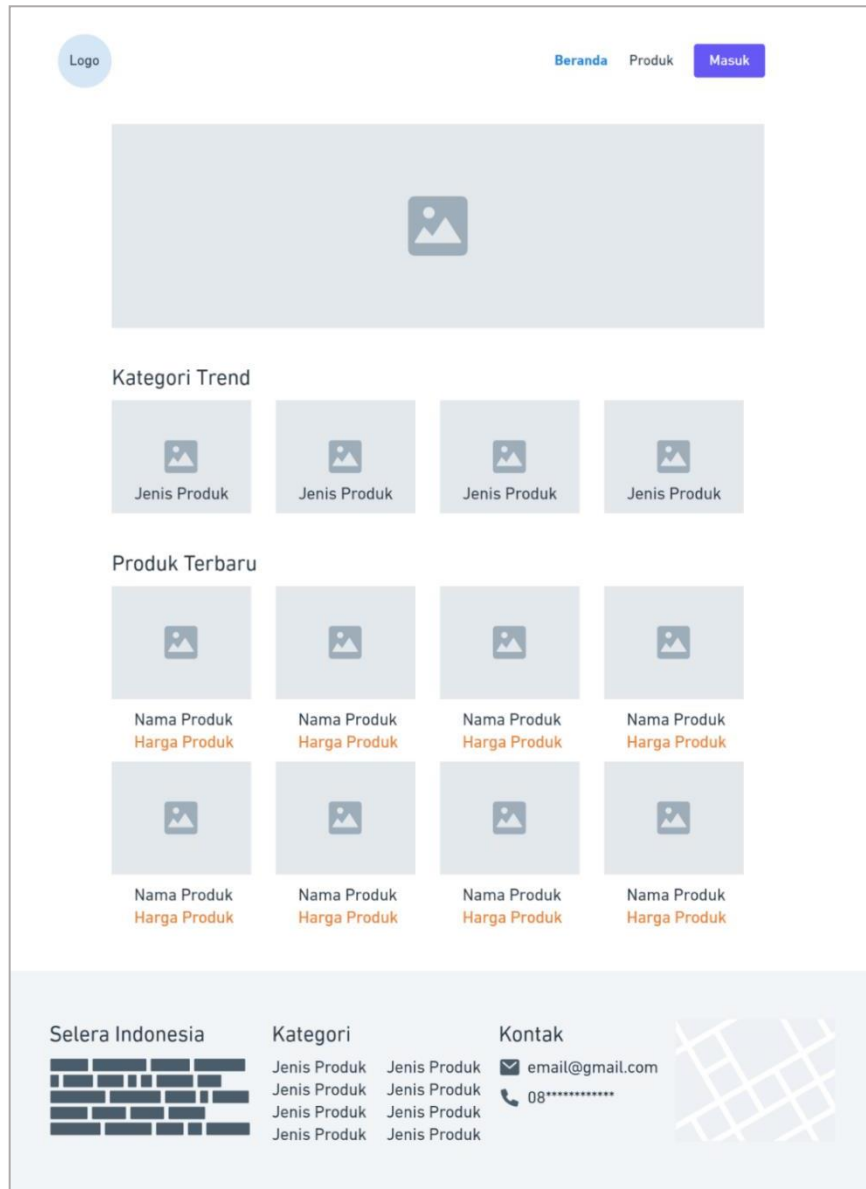


Gambar 3.1 Use Case Diagram

Berdasarkan gambar 3.1, *use case diagram* ini mengacu pada skenario penggunaan di mana seorang admin atau administrator sistem menggunakan sistem untuk melakukan tugas administratif atau manajemen sistem. Sementara itu, *use case diagram* pelanggan mengacu pada skenario penggunaan di mana seorang pelanggan menggunakan sistem untuk melakukan tugas-tugas seperti membuat akun, melakukan pembelian atau transaksi, hingga mengakses informasi produk atau layanan.

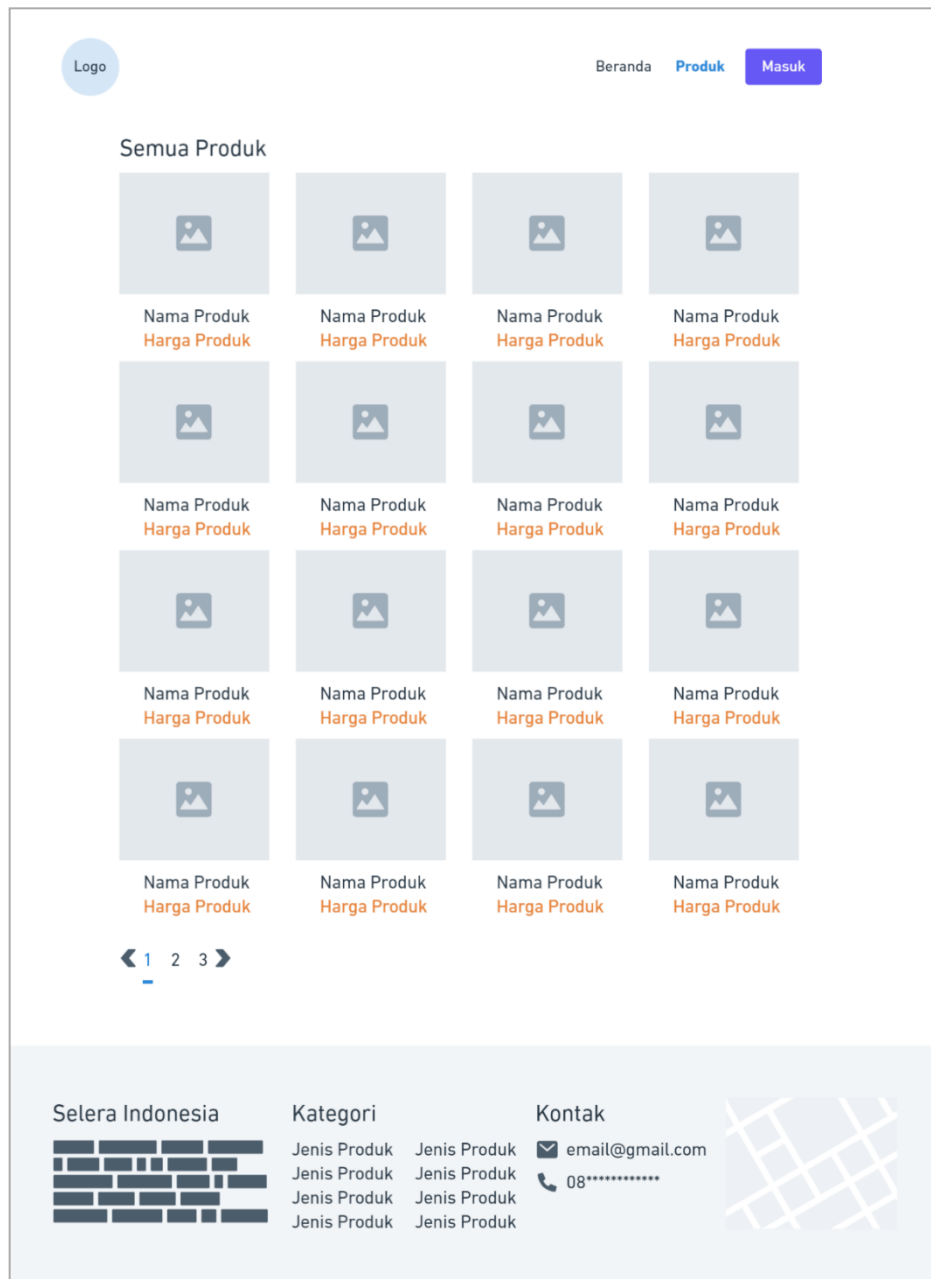
3.1.2 Desain Wireframe

Pada implementasi dalam bentuk *wireframe*, maka berikut ini tampilan sistem informasi pemasaran dan penjualan produk Selera Indonesia:



Gambar 3.2 Wireframe Halaman Beranda

Berdasarkan gambar 3.2 *wireframe* bagian halaman beranda, terdapat beberapa menu bar dan *highlight* yaitu produk terbaru dan kategori *trend* yang dapat menarik pengguna. Kemudian terdapat *footer* yang berisikan informasi mengenai Selera Indonesia.



Gambar 3.3 Wireframe Halaman Produk

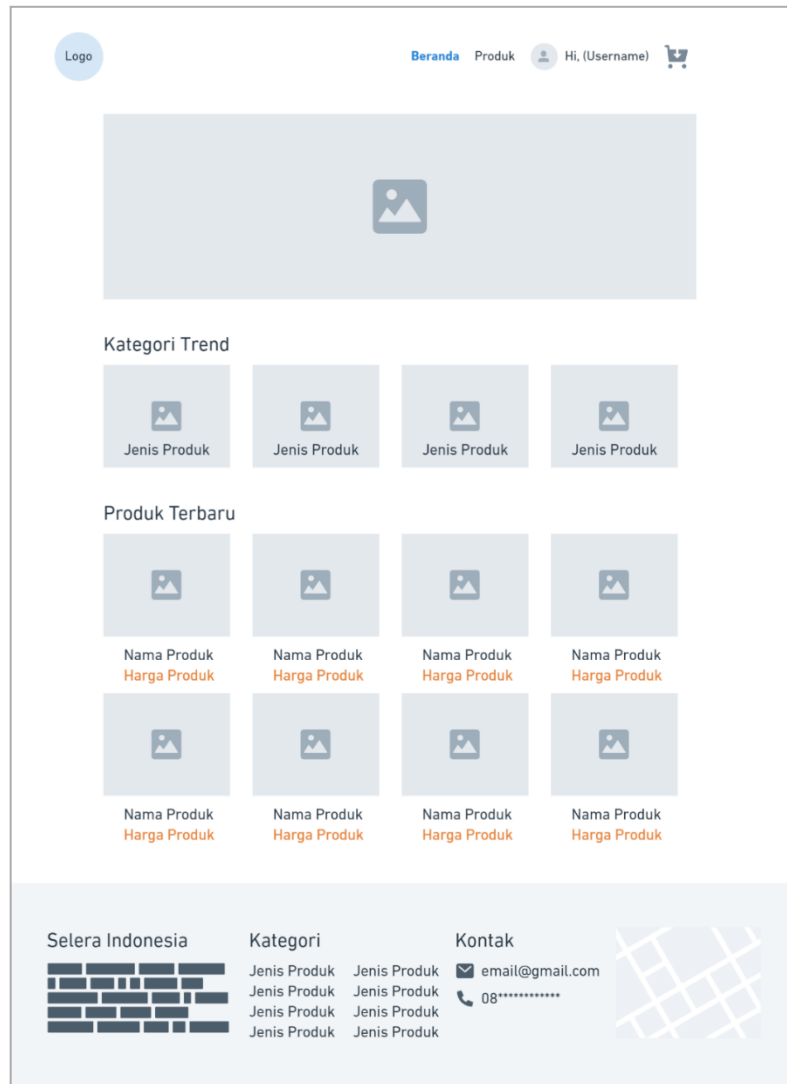
Berdasarkan gambar 3.3 *wireframe* bagian halaman produk, menampilkan semua produk yang dipasarkan oleh Selera Indonesia. Dalam menampilkan produk, terdapat nama produk dan harga produk yang dijual.

Gambar 3.4 *Wireframe* Halaman Akun *Login*

Berdasarkan gambar 3.4, terdapat *wireframe* halaman akun *login* untuk membuat pengguna dapat masuk ke dalam sistem dan mengakses fitur- fitur yang tersedia di dalamnya seperti melakukan transaksi. Pada halaman ini, pengguna dapat memasukkan *username* dan *password* yang telah terdaftar sebelumnya untuk melakukan proses autentikasi dan masuk ke dalam halaman utama sistem informasi Selera Indonesia.

Gambar 3.5 *Wireframe* Halaman Daftar Akun

Berdasarkan gambar 3.5, terdapat *wireframe* halaman daftar akun yang akan digunakan oleh pengguna untuk melakukan pendaftaran akun baru pada *website* Selera Indonesia. Halaman ini berisi *form* yang harus diisi oleh pengguna, seperti *username*, *email*, *password*, dan konfirmasi *password*. Setelah *form* diisi dengan benar, pengguna dapat menekan tombol "Daftar" untuk menyimpan informasi akun baru pada *database website* Selera Indonesia.



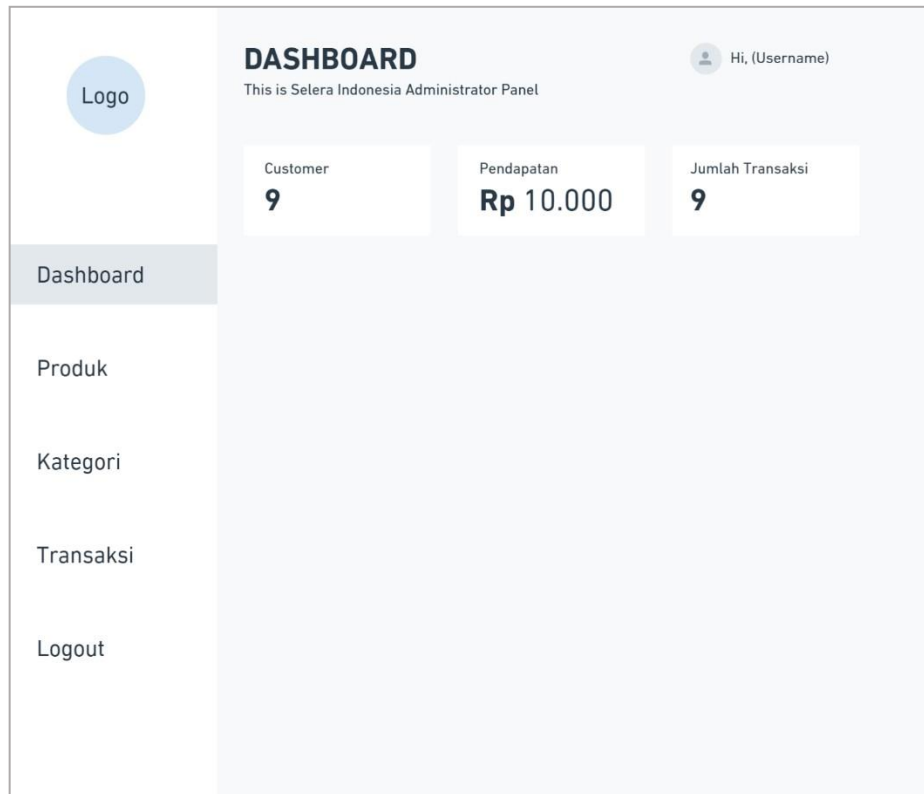
Gambar 3 6 Wireframe Halaman Beranda Akun Login

Berdasarkan gambar 3.6, pada *wireframe* halaman beranda sama seperti tampilan saat belum melakukan *login* akun. Yang berbeda saat *login* terdapat fitur *cart*/keranjang yang dapat digunakan oleh pengguna untuk menyimpan produk-produk yang ingin dibeli sehingga pengguna dapat memudahkan proses transaksi pembelian ketika melakukan *checkout*. Selain itu, pengguna juga dapat melihat riwayat transaksi yang pernah dilakukan pada halaman beranda akun *login* tersebut.

The wireframe illustrates a transaction page layout. At the top, there is a navigation bar with a 'Logo' placeholder, links for 'Beranda' and 'Produk', a user profile icon with the text 'Hi, (Username)', and a shopping cart icon. Below the navigation bar, a table displays a list of products. The table has four columns: 'Gambar', 'Produk', 'Kategori', and 'Harga'. Two product rows are shown, each with a placeholder image, the text 'Nama Produk', 'Jenis Kategori', and a price of 'Rp 9.999'. Under the product list, the 'Detail Pengiriman' section contains five input fields: 'Alamat', 'Kota', 'Provinsi', 'Kode Pos', and 'Nomor HP (WA)'. The 'Informasi Pembayaran' section at the bottom features a table with four columns: 'Subtotal Produk', 'Subtotal Pengiriman', 'Biaya Layanan', and 'Total Pembayaran'. The values are 'Rp 9.999', 'Rp 9.999', 'Rp 9.999', and 'Rp 9.999' respectively, with the total highlighted in blue. A 'Bayar Sekarang' button is positioned below the payment summary table.

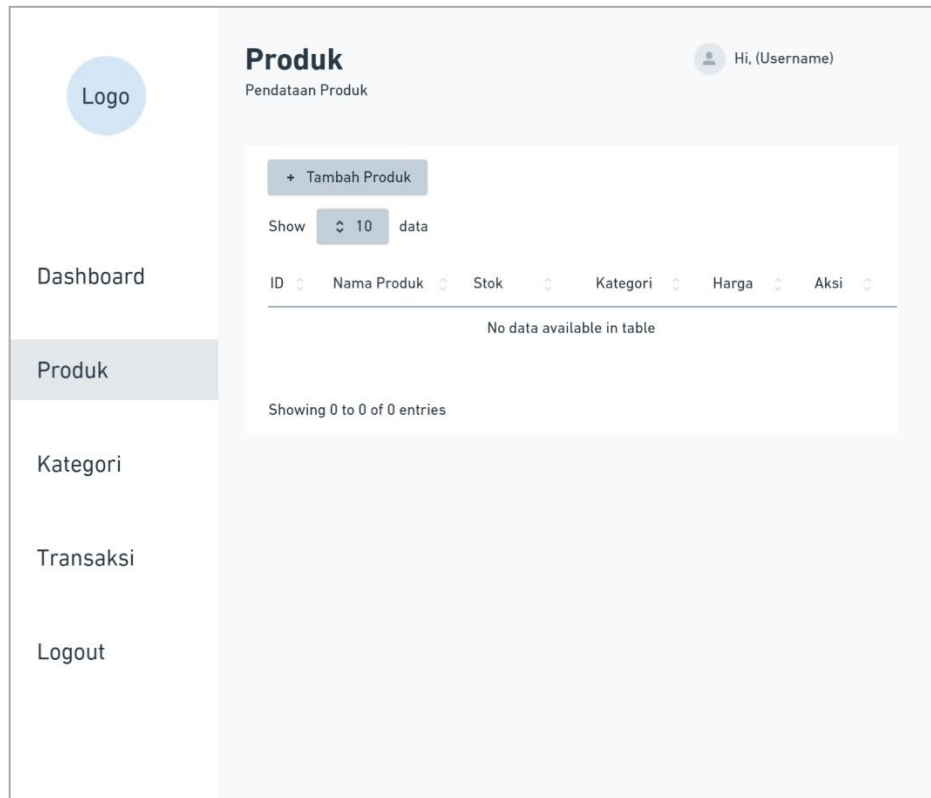
Gambar 3.7 *Wireframe* Halaman Transaksi

Berdasarkan gambar 3.7, pada *wireframe* halaman transaksi memiliki fungsi untuk melakukan transaksi. Pada halaman ini terdapat produk yang telah dimasukkan ke dalam keranjang, detail pengiriman untuk beberapa komponen seperti alamat, kota, provinsi, kode pos, dan nomor WA yang harus diisi oleh pengguna untuk dapat melakukan transaksi secara *online* pada *website* Selera Indonesia.



Gambar 3.8 Wireframe Halaman *Dashboard*

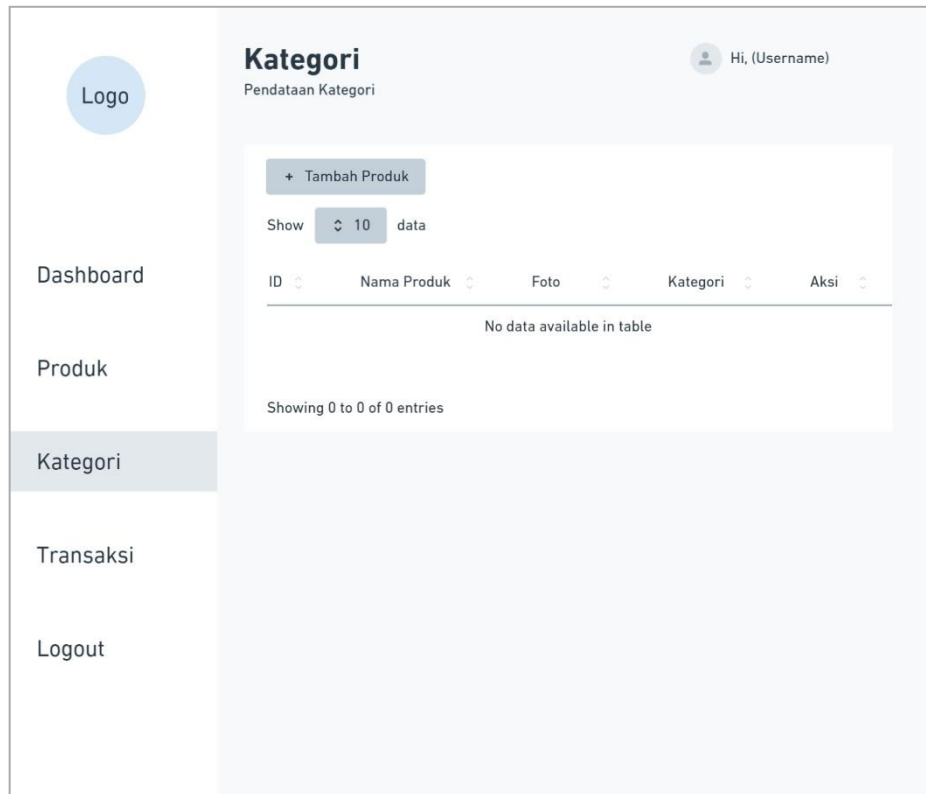
Berdasarkan gambar 3.8, *wireframe* halaman *dashboard* pada bagian *admin* untuk pemasaran dan penjualan produk merupakan halaman utama yang digunakan oleh administrator dalam mengelola data dan informasi terkait penjualan dan pemasaran produk pada *website* Selera Indonesia. Pada *dashboard* ini, administrator dapat melihat berbagai informasi seperti jumlah pengguna yang melakukan transaksi, jumlah pendapatan, dan jumlah transaksi yang berhasil.



Gambar 3.9 Wireframe Halaman Produk

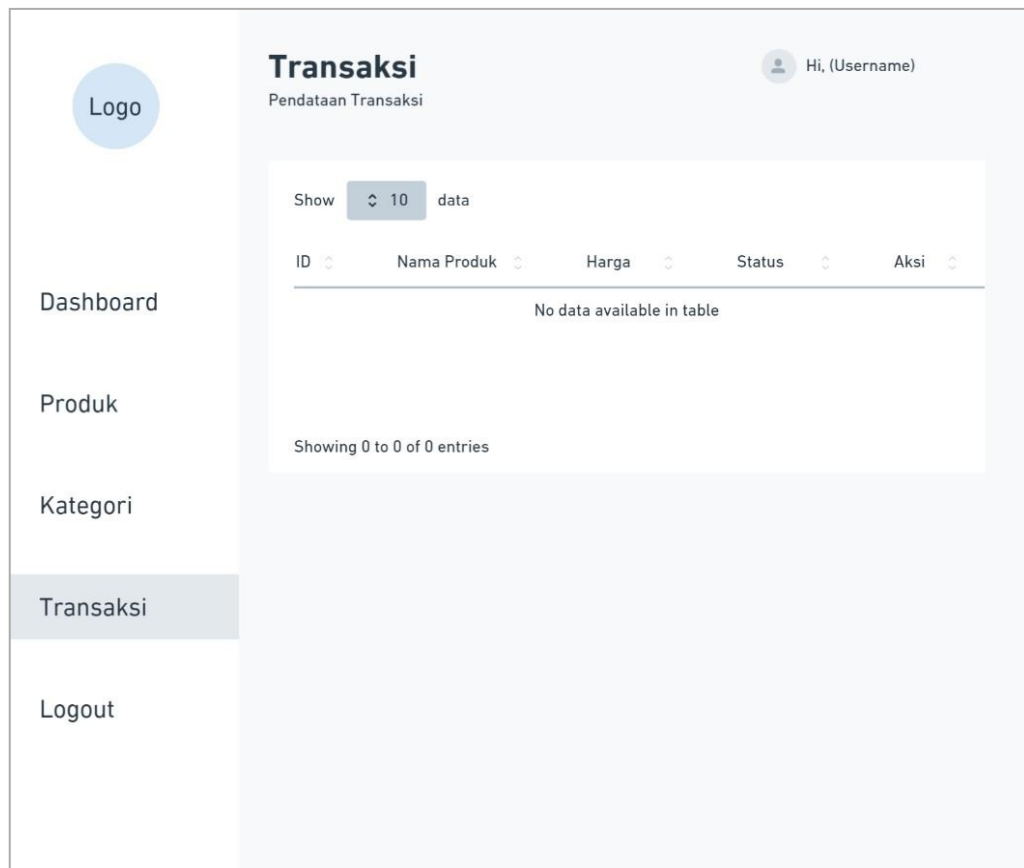
Berdasarkan gambar 3.9, *wireframe* halaman pada bagian admin merupakan halaman yang memuat informasi mengenai produk-produk yang tersedia pada toko *online*. Pada halaman ini, admin dapat melakukan manajemen terhadap produk seperti menambah, mengubah, atau menghapus produk. Informasi yang ditampilkan pada halaman ini meliputi nama produk, kategori produk, harga produk, dan aksi.

Admin juga dapat menambahkan gambar produk untuk memperjelas deskripsi produk dan memberikan gambaran yang lebih baik mengenai produk tersebut. Selain itu, pada halaman produk, admin juga dapat mengatur kategori produk. Halaman produk pada bagian admin sangat penting dalam pemasaran dan penjualan produk karena memuat informasi produk yang akan dilihat oleh pelanggan. Dengan informasi produk yang jelas dan lengkap, pelanggan dapat dengan mudah memilih produk yang diinginkan dan melakukan pembelian secara *online*.



Gambar 3.10 *Wireframe* Halaman Kategori

Berdasarkan gambar 3.10, admin dapat menambahkan, mengubah, atau menghapus kategori produk yang tersedia pada toko *online*. Kategori produk merupakan cara untuk mengelompokkan produk sesuai dengan jenisnya sehingga memudahkan pelanggan dalam mencari produk yang mereka butuhkan. Admin dapat menambahkan kategori baru jika kategori yang diinginkan tidak tersedia pada daftar kategori yang sudah ada. Pengelolaan kategori pada halaman *admin* ini sangat penting karena dapat membantu pengguna dalam menavigasi produk yang tersedia di *website*. Kategori produk yang jelas dan terstruktur dapat membantu pelanggan menemukan produk yang mereka cari dengan lebih mudah.



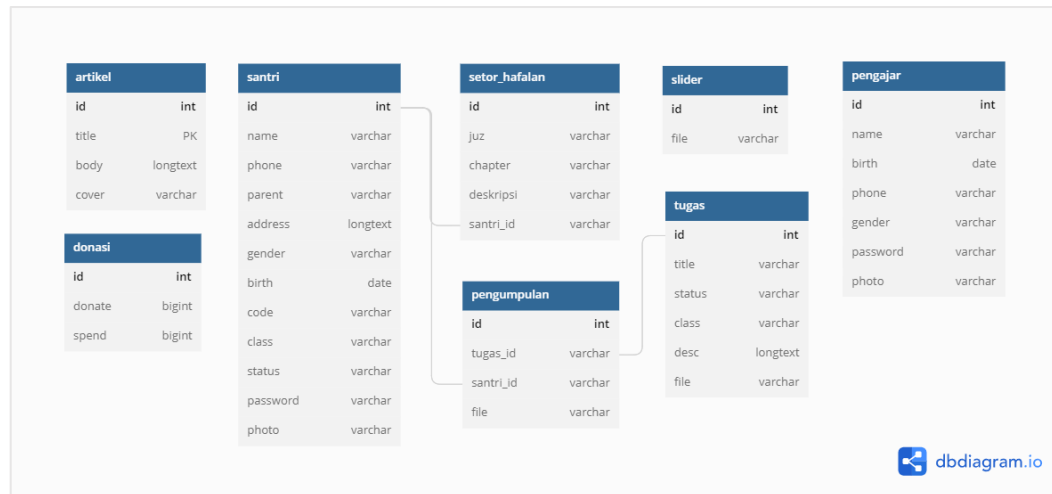
Gambar 3.11 *Wireframe* Halaman Transaksi

Berdasarkan gambar 3.11, transaksi pada bagian admin untuk pemasaran dan penjualan produk merupakan fitur yang sangat penting untuk mengelola dan memonitor semua transaksi yang terjadi pada *website* Selera Indonesia. Pada halaman transaksi di *dashboard* admin, terdapat daftar semua transaksi yang terjadi seperti pesanan, pembayaran, dan pengiriman barang. Informasi yang disajikan pada halaman transaksi antara lain nomor ID, nama produk, total harga, dan status transaksi. Fitur transaksi pada admin juga dilengkapi dengan fitur *filtering* dan *sorting*, sehingga memudahkan admin dalam mencari dan memonitor transaksi yang diinginkan.

3.1.3 Merancang Diagram *Database*

Setelah mengumpulkan semua data yang diperlukan untuk pengembangan situs *web* ini, langkah berikutnya adalah perancangan *database*. Untuk memudahkan proses ini, dipilih untuk membuat diagram terlebih dahulu. Dbdiagram.io digunakan sebagai *tool* pembuatan diagram *database*. *Tool* ini berbasis *online*,

dengan antarmuka dan tampilan yang intuitif, sehingga memudahkan pengguna dalam membuat diagram *database* sesuai kebutuhan. Diagram *database* untuk sistem informasi pemasaran dan penjualan Selera Indonesia dapat dilihat di bawah ini:



Gambar 3.12 *Diagram Database*

Dari diagram yang ditampilkan pada gambar 3.12, tabel-tabel yang diperlukan telah dibuat berdasarkan data yang telah dikumpulkan. Relasi antar tabel juga telah disusun agar saat perancangan *database* dilakukan nanti, tidak perlu dipertimbangkan ulang cara dan tabel apa yang perlu dihubungkan.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Antarmuka Pengguna

4.1.1 Implementasi Desain ke Dalam Proyek

Setelah membuat rancangan visual dan *database*, maka melanjutkan ke tahap pembuatan proyek Laravel nya beserta pembuatan tabel *migration* nya. Beberapa *tool* yang diperlukan untuk menginstal Laravel antara lain :

1. Composer
2. Node Package Manager
3. Text Editor

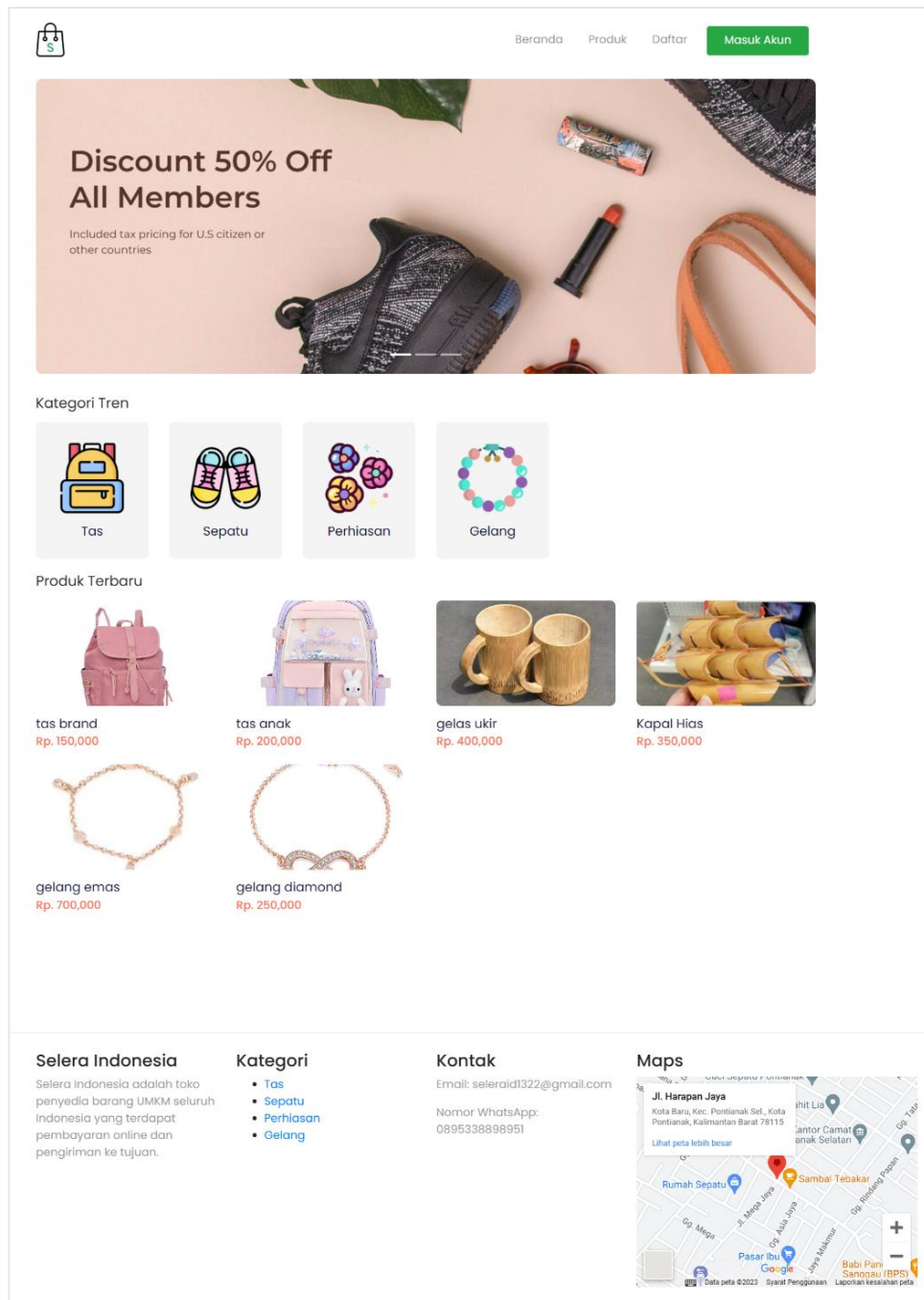
Tool-tool tersebut memiliki fungsi nya masing-masing. *Composer* dimanfaatkan sebagai dependensi untuk mempermudah pengelolaan instalasi dan pembaruan pustaka di dalam Laravel dengan lebih efisien. Node Package Manager atau yang lebih dikenal sebagai NPM, digunakan sebagai alat untuk memudahkan pengelolaan paket di dalam Laravel. Terakhir, editor teks, yang dipergunakan untuk mengatur berbagai jenis kode atau berkas proyek yang dimaksudkan untuk digunakan dalam Laravel, adalah Visual Studio Code, seperti yang dipilih oleh penulis. Setelah semua alat ini berhasil diinstal, instalasi Laravel dapat dilakukan dengan lancar untuk memulai proyek.

Dalam kerangka Laravel, terdapat konsep arsitektur pengembangan aplikasi yang dikenal sebagai MVC (*Model, View, Controller*), yang telah dibahas sebelumnya. Setelah merencanakan semua tampilan yang diperlukan, langkah berikutnya melibatkan pengaturan migrasi tabel data ke dalam database yang terhubung. Konsep arsitektur ini dalam Laravel memberikan kemudahan pengguna berkat struktur proyek yang diorganisir dengan baik. Dalam proses migrasi yang telah disediakan oleh Laravel, dilakukan pengisian bidang-bidang yang dibutuhkan di dalam file migrasi. Laravel juga menyediakan perintah-perintah seperti migrasi, model, atau kontrol melalui command prompt yang dapat diakses melalui Visual Studio Code. Dengan pendekatan ini, setiap kali diperlukan migrasi tabel baru,

cukup menjalankan perintah yang sesuai, dan Laravel secara otomatis akan membuat file migrasi yang dibutuhkan.

Proses awal dalam mengembangkan proyek *website* dimulai dengan mengatur koneksi ke *database* yang telah ada. Setelah itu, perhatian dipindahkan ke perancangan tata letak halaman menggunakan kode dalam Laravel, yang dikenal sebagai *View*. Dalam kerangka Laravel, terdapat direktori "*view*" yang digunakan untuk mengatur berbagai halaman sesuai kebutuhan, sejalan dengan data yang telah disiapkan pada tahap desain awal. Tata letak dasar yang telah dibuat untuk setiap halaman dalam direktori "*view*" mengikuti rencana awal yang sudah disusun. Dari sudut pandang ini, terlihat jelas bahwa pembuatan kerangka dasar atau gambaran awal aplikasi secara signifikan mempermudah proses pengembangan proyek. Berikut adalah beberapa hasil tampilan yang telah dibuat berdasarkan tata letak *wireframe* :

1. Tampilan Halaman Beranda



Gambar 4.1 Tampilan Halaman Beranda

Berdasarkan gambar 4.1, menampilkan tampilan halaman dengan navigasi bar yang muncul, menyediakan akses ke berbagai kategori dan produk yang dijual oleh Selera Indonesia. Di bawah ini adalah kode yang merangkum isi dari *Controller* halaman beranda tersebut:

```
<?php

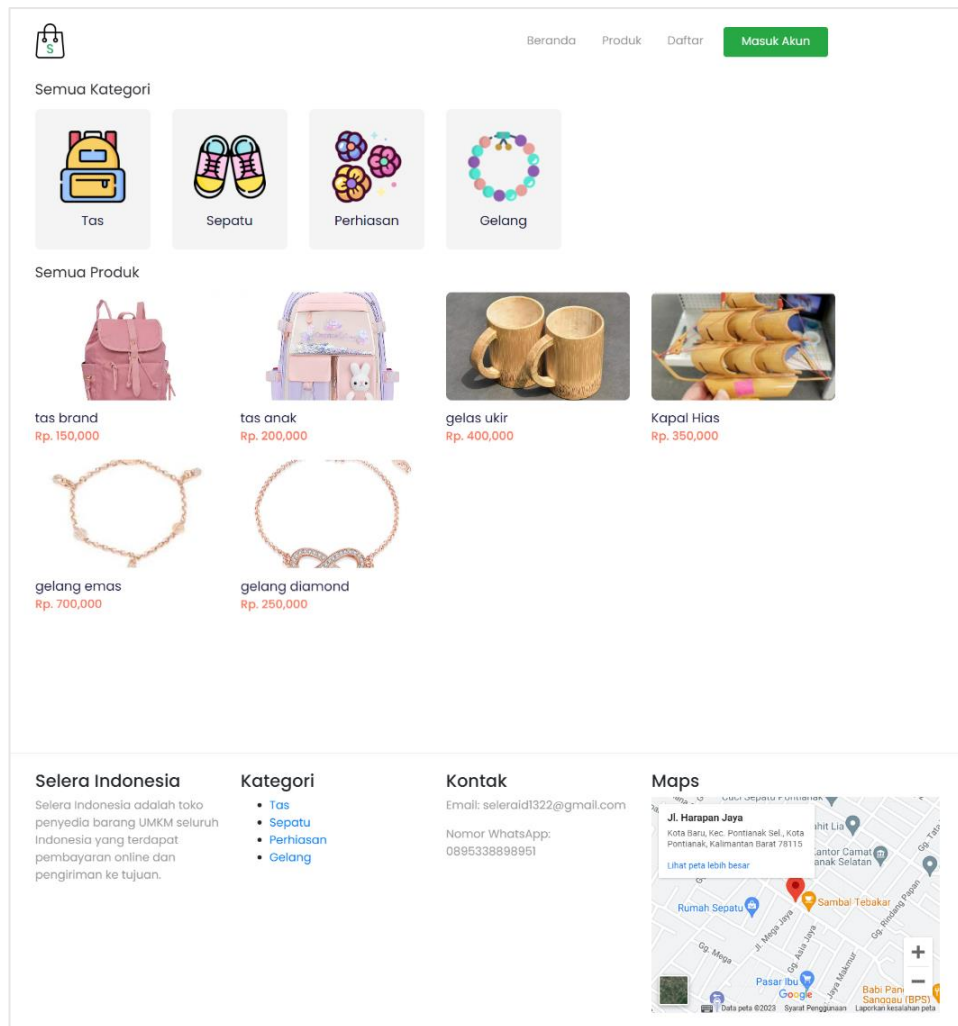
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Category;
use App\Product;

class HomeController extends Controller
{
    public function index()
    {
        $categories = Category::take(6)->get();
        $products = Product::with('galleries')->take(8)->get();

        return view('pages.home',[
            'categories' => $categories,
            'products' => $products
        ]);
    }
}
```

2. Tampilan Halaman Produk



Gambar 4.2 Tampilan Halaman Produk

Berdasarkan gambar 4.2, menggambarkan halaman produk yang berperan dalam menampilkan konten produk dan juga mengatur penyebaran dalam kategori. Dengan demikian, di bawah ini terdapat kode yang berfungsi dalam *Controller*:

```
<?php

namespace App\Http\Controllers;

use App\Category;
use App\Product;

use Illuminate\Http\Request;

class CategoryController extends Controller
```

```
{

    public function index(Request $request)
    {
        $categories = Category::all();
        $products = Product::paginate($request-
>input('limit', 12));

        return view('pages.category',[
            'categories' => $categories,
            'products' => $products
        ]);
    }

    public function detail(Request $request, $slug)
    {
        $categories = Category::all();
        $category = Category::where('slug', $slug)-
>firstOrFail();
        $products = Product::where('categories_id',
        $category->id)->paginate($request->input('limit', 12));

        return view('pages.category',[
            'categories' => $categories,
            'category' => $category,
            'products' => $products
        ]);
    }
}
```


3. Tampilan Halaman *Login*

Gambar 4.3 Tampilan Halaman *Login*

Berdasarkan gambar 4.3, halaman ini menyajikan elemen untuk *login*, di mana pengguna diminta untuk memasukkan alamat *email* dan kata sandi. Untuk mengakses konfigurasi data yang diperlukan, berikut ini adalah kode yang ada dalam *Controller*:

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
```

```
{  
    use AuthenticatesUsers;  
    protected function redirectTo() {  
        if (Auth::user()->roles === 'ADMIN') {  
            return RouteServiceProvider::ADMIN;  
        } else {  
            return RouteServiceProvider::HOME;  
        }  
    }  
  
    public function __construct()  
    {  
        $this->middleware('guest')->except('logout');  
    }  
}
```

4. Tampilan Halaman Daftar Akun

Gambar 4.4 Tampilan Halaman Daftar Akun

Berdasarkan gambar 4.4, menggambarkan halaman yang ditujukan bagi pengguna untuk melakukan pendaftaran, yang diperlukan agar mereka dapat melaksanakan transaksi. Oleh karena itu, berikut ini terdapat implementasi kode pada *Controller* halaman pendaftaran akun:

```
<?php

namespace App\Http\Controllers\Auth;

use App\User;
```

```

use App\Category;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use App\Providers\RouteServiceProvider;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
    use RegistersUsers;
    protected $redirectTo = RouteServiceProvider::HOME;
    public function __construct()
    {
        $this->middleware('guest');
    }

    public function showRegistrationForm()
    {
        $categories = Category::all();
        return view('auth.register', [
            'categories' => $categories
        ]);
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email',
                'max:255', 'unique:users'],
            'password' => ['required', 'string',
                'min:8', 'confirmed'],
            'store_name' => ['nullable', 'string',
                'max:255'],
            'categories_id' => ['nullable', 'integer',
                'exists:categories,id'],
            'is_store_open' => ['required'],
            'photo' => ['nullable', 'image',
                'max:2048'],
        ]);
    }

    protected function create(array $data)
    {
        $datas = $data['photo']->store('assets/user',
            'public');
    }
}

```

```

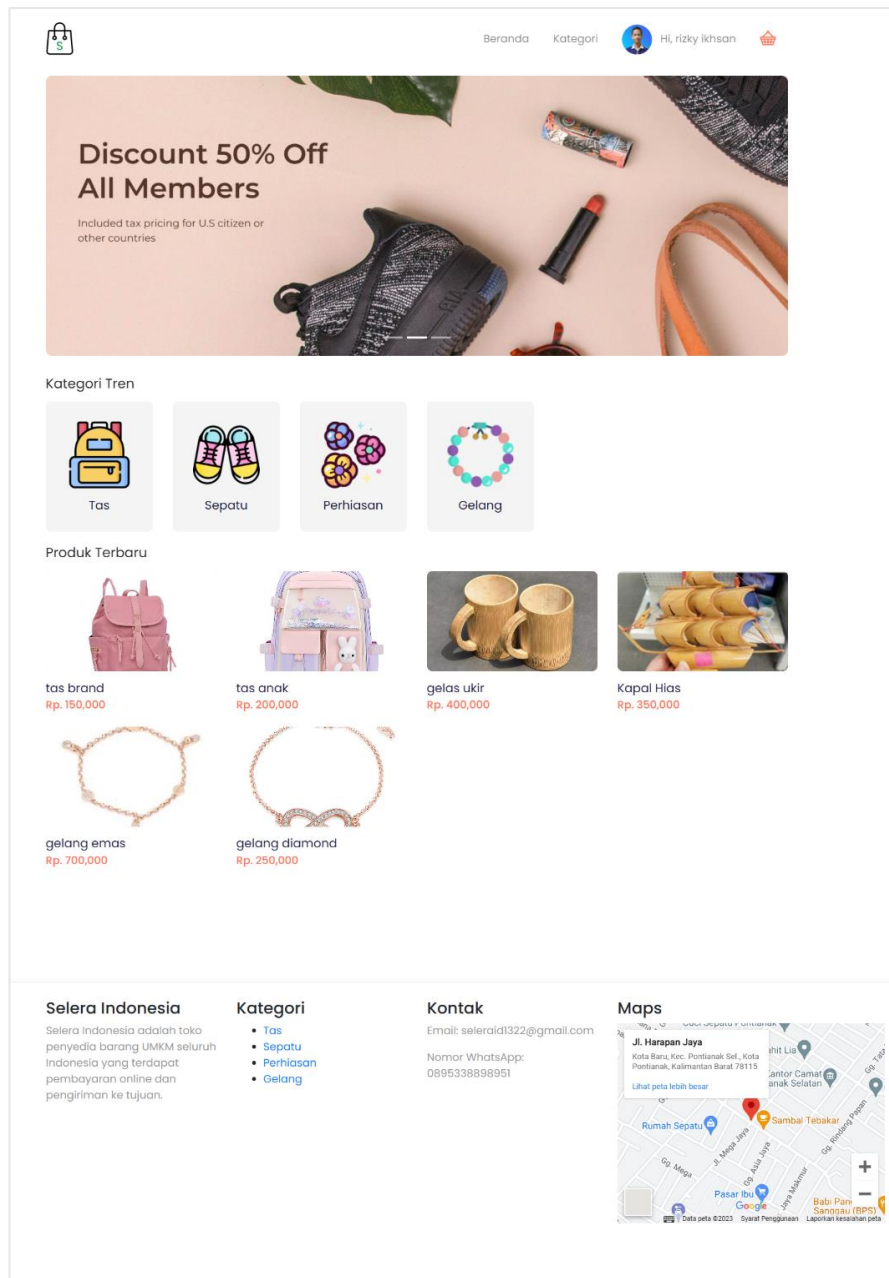
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
            'store_name' => isset($data['store_name']) ?
$data['store_name'] : '',
            'categories_id' =>
isset($data['categories_id']) ? $data['categories_id'] :
NULL,
            'store_status' => $data['is_store_open'] ? 1
: 0,
            'photo' => $datas
        ]);
    }

    public function success()
    {
        return view('auth.success');
    }

    public function check(Request $request)
    {
        return User::where('email', $request->email)-
>count() > 0 ? 'Unavailable' : "Available" ;
    }
}

```

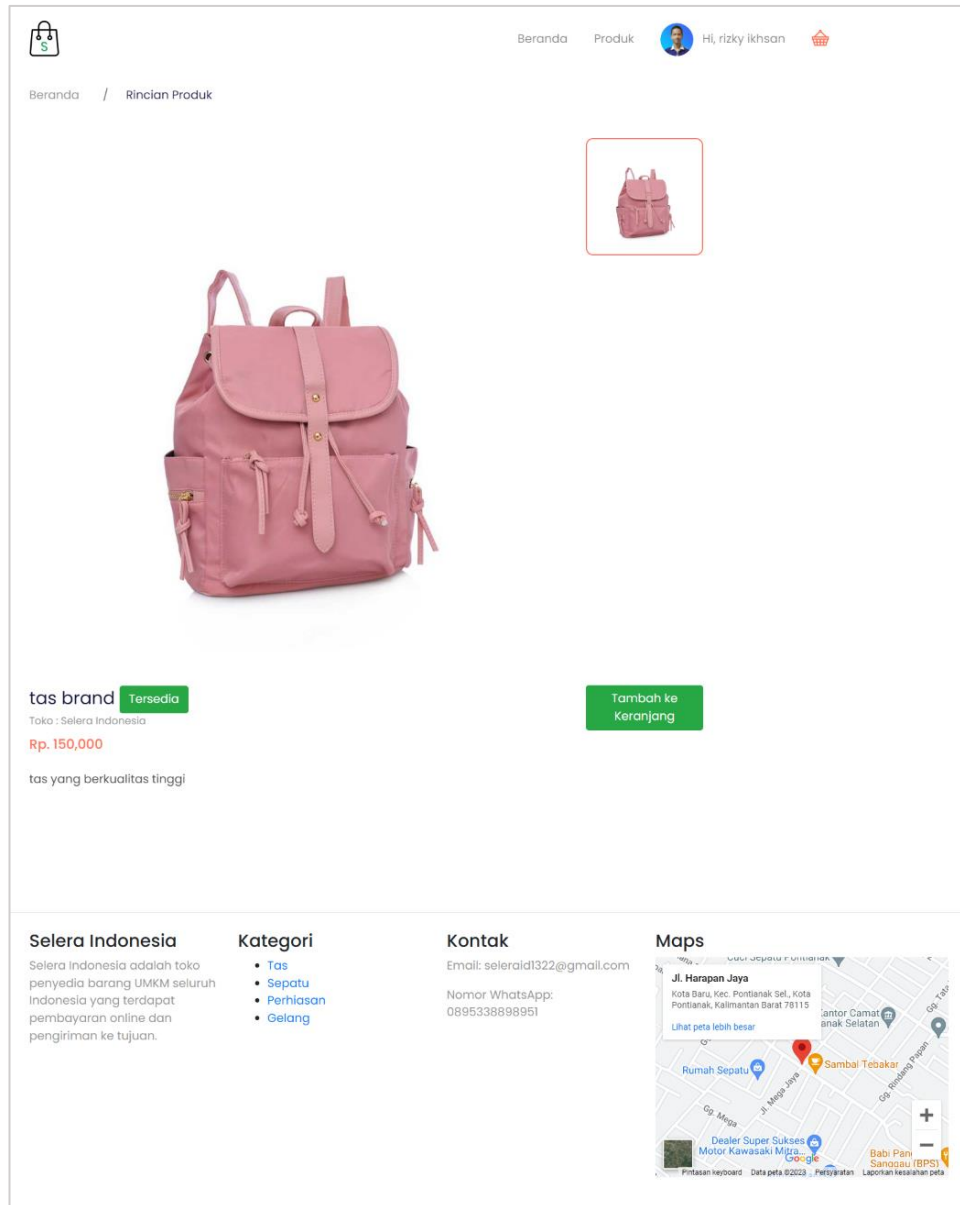
5. Tampilan Halaman Beranda Akun *Login*



Gambar 4.5 Tampilan Halaman Beranda Akun *Login*

Berdasarkan gambar 4.5, halaman beranda setelah *login* menampilkan tampilan beranda sebelumnya, namun ditambahkan dengan foto profil pengguna yang telah berhasil masuk ke dalam akunnya.

6. Tampilan Rincian Produk



Gambar 4.6 Tampilan Rincian Produk

Berdasarkan gambar 4.6, halaman rincian produk yang menampilkan tampilan produk, harga produk, ketersediaan produk serta tombol tambah ke keranjang. berikut hasil implementasi kode yang terdapat dalam *Blade*:

```
@extends('layouts.app')
```

```

@section('title')
    Halaman Keranjang
@endsection

@section('content')

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/boo
tstrap.min.css" rel="stylesheet"
    integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCheKRQNPtmoHDEXuppvndJzQIu
9" crossorigin="anonymous">
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/css/select2.min.css" rel="stylesheet" />
    <!-- Page Content -->
    <div class="page-content page-cart">
        <section class="store-breadcrumbs" data-aos="fade-down"
data-aos-delay="100">
            <div class="container">
                <div class="row">
                    <div class="col-12">
                        <nav>
                            <ol class="breadcrumb">
                                <li class="breadcrumb-item">
                                    <a
href="/index.html">Beranda</a>
                                </li>
                                <li class="breadcrumb-item
active">
                                    Keranjang
                                </li>
                            </ol>
                        </nav>
                    </div>
                </div>
            </div>
        </section>

        <section class="store-cart">
            <div class="container">
                <div class="row" data-aos="fade-up" data-aos-
delay="100">
                    <div class="col-12 table-responsive">
                        <table class="table table-borderless
table-cart">
                            <thead>

```



```
<tr>  
    <td>Gambar</td>  
    <td>>Nama &amp; Penjual</td>  
    <td>Harga</td>  
    <td>Jumlah</td>  
    <td>Menu</td>  
</tr>  
</thead>  
<tbody>  
    @php $totalPrice = 0 @endphp  
    @foreach ($carts as $cart)  
        <tr>  
            <td style="width:  
20%;">  
  
                @if ($cart->  
>product->galleries)  
  
                      
  
                @endif  
            </td>  
            <td style="width:  
20%;">  
  
                <div  
class="product-title">{{ $cart->product->name }}</div>  
                <div  
class="product-subtitle">by {{ $cart->product->user->store_name  
}}</div>  
  
            </td>  
            <td style="width:  
15%;">  
  
                <div  
class="product-title">Rp. {{ number_format($cart->product->  
>price) }}</div>  
  
                <div  
class="product-subtitle">Rupiah</div>  
  
            </td>  
            <td style="width:  
25%;">  
  
                <div class="input-  
group" style="max-width: 150px;">  
  
                    <input  
type="number" name="amount" id="amount" class="form-control"  
value="1">  
  
                <div  
class="input-group-append">
```

```

<button
class="btn btn-outline-secondary" type="button"
onclick="decrementAmount()">-</button>

<button
class="btn btn-outline-secondary" type="button"
onclick="incrementAmount()">+</button>

</div>
</div>
</td>
<td style="width:
20%;">
<form action="{route('cart-delete', $cart->id) }}" method="POST">
@method('DELETE
')
@csrf
<button
class="btn btn-remove-cart" type="submit">
Hapus
</button>
</form>
</td>
</tr>
@php
$allPrice = 0;
$allPrice += $cart-
>product->price;

@endphp
@php
$totalPrice += $cart-
>product->price ;

@endphp
@endforeach
</tbody>
</table>
</div>
</div>
<div class="row" data-aos="fade-up" data-aos-
delay="150">
<div class="col-12">
<hr />
</div>
<div class="col-12">
<h2 class="mb-4">Rincian Belanja</h2>
</div>
</div>
<main class="py-3">

```

```

<div class="container">
  <div class="row">
    <div class="col-12">
      <h2 class="fs-5 py-4 text-
center">
        Cek Ongkos Kirim
      </h2>
      <div class="card border rounded
shadow">
        <div class="card-body">
          <form id="form">
            <p class="bold
text-center">Paket Dikirim Dari Pontianak Kalimantan Barat
            </p>
            <div class="row mb-
3">
              <strong>Alamat
Tujuan</strong>
              <div
class="col-md-6">
                <label
for="destination_province" class="form-label">Provinsi</label>
                <select
name="destination_province" id="destination_province"
class="
form-select">
                  <option
>Pilih Provinsi</option>
                </select>
              </div>
              <div
class="col-md-6">
                <label
for="destination_city" class="form-label">Kota</label>
                <select
name="destination_city" id="destination_city"
class="
form-select">
                  <option
>Pilih Kota</option>
                </select>
              </div>
            </div>
            <div class="row mb-
3">
              <div
class="col-md-6">

```

```

<label
for="courier" class="form-label">Kurir</label>
<select
name="courier" id="courier" class="form-select">
<option
>Pilih Kurir</option>
<option
value="jne">JNE</option>
<option
value="pos">POS</option>
<option
value="tiki">TIKI</option>
</select>
</div>
<div
class="col-md-6">
<label
for="weight" class="form-label">Berat (Gram)</label>
<input
disabled type="number" name="weight" id="weight"
value="
1000" class="form-control">
</div>
</div>
<div class="col-
12">
<button
class="btn btn-primary" id="checkBtn">Cek</button>
</div>
</form>
</div>
</div>
<div id="result" class="mt-3 d-
none"></div>
</div>
</div>
</div>
</main>
<form action="{{ route('checkout') }}"
method="POST" enctype="multipart/form-data">
@csrf
@php
$totalPrice += $ongkir;
@endphp
<input type="hidden" name="total_price"
value="{{ $totalPrice }}">

```

```

        </div>
        <div class="row" data-aos="fade-up" data-
aos-delay="150">
            <div class="col-12">
                <hr />
            </div>
            <div class="col-12">
                <h2 class="mb-1">Informasi
Pembayaran</h2>
                @if ($ongkir != 0)
                    <p>Harga Total : Rp.{{
number_format($totalPrice ?? 0) }} (telah termasuk ongkir)
                    </p>
                @else
                    <p class="text-danger">Harga
Total : Rp.{{ number_format($totalPrice ?? 0) }} (Harap Cek
                    Ongkir Terlebih Dahulu)
                    </p>
                @endif

            </div>
        </div>
        <div class="row" data-aos="fade-up" data-
aos-delay="200">
            <div class="col-4 col-md-2">
                <div class="product-title text-
success">Rp.{{ number_format($totalPrice ?? 0) }}</div>
                <div class="product-
subtitle">Total</div>
            </div>
            <div class="col-8 col-md-3">
                <button {{ !$ongkir ? 'disabled' :
'' }} type="submit"
                    class="btn btn-success mt-4 px-
4 btn-block">
                    Pesan Sekarang
                </button>
            </div>
        </div>
    </form>
</div>
</section>
</div>
@endsection

```

```

@push('addon-script')
  <script src="/vendor/vue/vue.js"></script>
  <script src="https://unpkg.com/vue-toasted"></script>
  <script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
    <script src="https://code.jquery.com/jquery-3.7.0.min.js"
        integrity="sha256-
2Pmvv0kuTBOenSvLm6bvfbSSHrUJ+3A7x6P5Ebd07/g="
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/boots
trap.bundle.min.js"
        integrity="sha384-
HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrk
m" crossorigin="anonymous">
    </script>
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/js/select2.min.js"></script>
    <script>
        $(document).ready(function() {
            $('#origin_province,
#destination_province').select2({
                ajax: {
                    url: "{{ route('provinces') }}",
                    type: 'GET',
                    dataType: 'json',
                    delay: 250,
                    data: function(params) {
                        return {
                            keyword: params.term
                        }
                    },
                    processResults: function(response) {
                        return {
                            results: response
                        }
                    },
                },
                // Set nilai default untuk origin_province
                berdasarkan data dari route('provinces')
                initSelection: function(element, callback) {
                    var defaultProvinceId = 12; // Ganti dengan
                    nilai default yang Anda inginkan
                    var defaultProvince = null;
                    $.ajax({
                        url: "{{ route('provinces') }}",
                        dataType: 'json',

```

```

                success: function(response) {
                    defaultProvince =
response.find(function(province) {
                        return province.id ===
defaultProvinceId;
                    });
                    if (defaultProvince) {
                        callback(defaultProvince);
                    }
                });
            }
        });

        $('#origin_city, #destination_city').select2();

        $('#origin_province').on('change', function() {
            $('#origin_city').empty();
            $('#origin_city').append('<option>Choose
City</option>');
            $('#origin_city').select2('close');
            $('#origin_city').select2({
                ajax: {
                    url: "{ route('cities') }",
                    type: 'GET',
                    dataType: 'json',
                    delay: 250,
                    data: function(params) {
                        return {
                            keyword: params.term,
                            province_id:
$('#origin_province').val()
                        }
                    },
                    processResults: function(response) {
                        return {
                            results: response
                        }
                    },
                }
            });
        });

        $('#destination_province').on('change', function()
{
            $('#destination_city').empty();

```

```

        $('#destination_city').append('<option>Choose
City</option>');
        $('#destination_city').select2('close');
        $('#destination_city').select2({
            ajax: {
                url: "{{ route('cities') }}",
                type: 'GET',
                dataType: 'json',
                delay: 250,
                data: function(params) {
                    return {
                        keyword: params.term,
                        province_id:
$('#destination_province').val()
                    }
                },
                processResults: function(response) {
                    return {
                        results: response
                    }
                },
            }
        });
    });

    $('#checkBtn').on('click', function(e) {
        e.preventDefault();
        let origin = $('#origin_city').val();
        let destination = $('#destination_city').val();
        let courier = $('#courier').val();
        let weight = $('#weight').val();
        $.ajax({
            url: "{{ route('check-ongkir') }}",
            type: 'POST',
            dataType: 'json',
            data: {
                _token: "{{ csrf_token() }}",
                origin: origin,
                destination: destination,
                courier: courier,
                weight: weight
            },
            beforeSend: function() {
                $('#checkBtn').html('Loading...');
                $('#checkBtn').attr('disabled', true);
            },
            success: function(response) {

```



```

$('#result').removeClass('d-none');
$('#checkBtn').html('Submit');
$('#checkBtn').attr('disabled', false);
$('#result').empty();
$('#result').append(`
    <div class="col-12">
        <div class="card border
rounded shadow">
            <div class="card-body">
                <table class="table
table-bordered">
                    <thead>
                        <tr>
                            <th>Ser
vice</th>
                            <th>Des
cription</th>
                            <th>Har
ga</th>
                            <th>Est
imasi kedatangan</th>
                            <th>Pil
ihan</th>
                        </tr>
                    </thead>
                    <tbody
id="resultBody">
                        <tbody>
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
`);
$.each(response, function(i, val) {
    $('#resultBody').append(`
        <tr>
            <td>${val.service}</td>
            <td>${val.description}</td>
            <td>${val.cost[0].value}</td>
            <td>${val.cost[0].etd}</td>
            <td>
                <button class="btn btn-primary"
onclick="redirectToCart(${val.cost[0].value})">Pilih</button>
            </td>
        </tr>
    `);
});

```

```

        },
        error: function(xhr) {
            console.log(xhr.responseText);
        }
    });
});
});

function redirectToCart(value) {
    console.log('Value:', value);
    window.location.href = "{{ route('cart')
}}?ongkir=" + value;
}
</script>
<script>
var locations = new Vue({
    el: "#locations",
    mounted() {
        this.getProvincesData();
    },
    data: {
        provinces: null,
        regencies: null,
        provinces_id: null,
        regencies_id: null,
    },
    methods: {
        getProvincesData() {
            var self = this;
            axios.get('{{ route('api-provinces') }}')
                .then(function(response) {
                    self.provinces = response.data;
                })
        },
        getRegenciesData() {
            var self = this;
            axios.get('{{ url('api/regencies') }}/' +
self.provinces_id)
                .then(function(response) {
                    self.regencies = response.data;
                })
        },
    },
    watch: {

```

```

        provinces_id: function(val, oldVal) {
            this.regencies_id = null;
            this.getRegenciesData();
        },
    },
    });
</script>
<script>
    function incrementAmount() {
        var amountInput =
document.getElementById('amount');
        amountInput.value = parseInt(amountInput.value) +
1;
    }

    function decrementAmount() {
        var amountInput =
document.getElementById('amount');
        var currentAmount = parseInt(amountInput.value);
        if (currentAmount > 1) {
            amountInput.value = currentAmount - 1;
        }
    }
</script>

@endpush

```

7. Tampilan Halaman Transaksi

Beranda / Keranjang

Gambar	Nama & Penjual	Harga	Menu
	gelang emas by Selera Indonesia	Rp. 700,000 Rupiah	Hapus
	gelas ukir by Selera Indonesia	Rp. 400,000 Rupiah	Hapus
	Kapal Hias by Selera Indonesia	Rp. 350,000 Rupiah	Hapus

Rincian Belanja

Cek Ongkos Kirim

Paket Dikirim Dari Pontianak Kalimantan Barat

Alamat Tujuan
 Province: City:
 Courier: Weight (Gram):

Alamat 1:
 Alamat 2:
 Provinsi: Kota: Kode Pos:
 Negara: No. WhatsApp:

Informasi Pembayaran
 Harga Total : Rp.1,450,000 (Harap Cek Ongkir Terlebih Dahulu)
 Rp.1,450,000
 Total

Selera Indonesia
 Selera Indonesia adalah toko penyedia barang UMKM seluruh Indonesia yang terdapat pembayaran online dan pengiriman ke tujuan.

Kategori
 • Tas
 • Sepatu
 • Perhiasan
 • Gelang

Kontak
 Email: seleraid1322@gmail.com
 Nomor WhatsApp: 0895338898951

Maps
 Jl. Hergan Jaya
 Kota Bang, Kecamatan Pontianak Sel., Kota Pontianak, Kalimantan Barat 78115
 Lihat peta lebih besar

Gambar 4.7 Tampilan Halaman Transaksi

Berdasarkan gambar 4.7, menggambarkan halaman transaksi yang memungkinkan pelaksanaan transaksi, termasuk langkah-langkah seperti memasukkan informasi alamat untuk menghitung harga ongkos kirim dari

Raja Ongkir. Sehubungan dengan ini, berikut hasil implementasi kode yang terdapat dalam *Controller*:

```
<?php

namespace App\Http\Controllers;
use App\Cart;
use App\Category;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

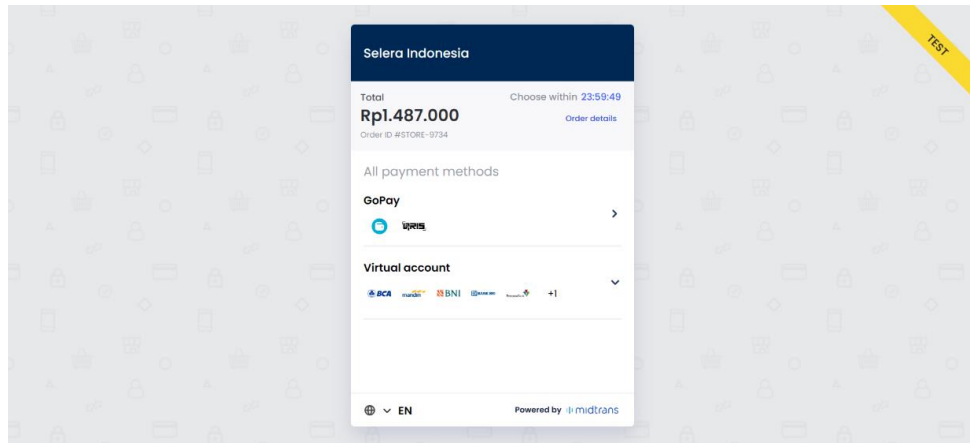
class CartController extends Controller
{
    public function index(Request $request)
    {
        $tujuan = Auth::user()->regencies_id;
        $categories = Category::take(6)->get();
        $ongkir = $request->query('ongkir');
        $carts = Cart::with(['product.galleries',
        'user'])->where('users_id', Auth::user()->id)-
        >get();

        return view('pages.cart', [
            'carts' => $carts,
            'categories' => $categories,
            'ongkir' => $ongkir
        ]);
    }

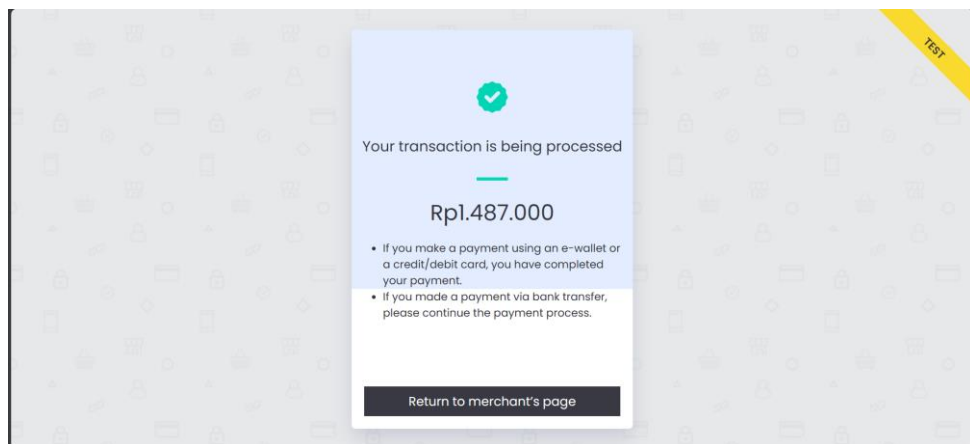
    public function delete(Request $request, $id)
    {
        $cart = Cart::findOrFail($id);
        $cart->delete();
        return redirect()->route('cart');
    }

    public function success()
    {
        return view('pages.success');
    }
}
```

Kemudian setelah menginputkan semua data dan klik tombol “pesan sekarang” maka akan memunculkan tampilan dari API Midtrans untuk melakukan pembayaran, maka berikut tampilan dari Midtrans:



Gambar 4.8 Menampilkan Total Pembayaran di Midtrans



Gambar 4.9 Tampilan Ketika Transaksi Berhasil

Berdasarkan gambar 4.8 dan 4.9, berikut hasil implementasi kode yang terdapat dalam *Controller*:

```
<?php

namespace App\Http\Controllers;

use App\Cart;
use Exception;
use Midtrans\Snap;
use App\Transaction;
use Midtrans\Config;
use App\TransactionDetail;
```

```

use Midtrans\Notification;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class CheckoutController extends Controller
{
    public function process(Request $request)
    {
        $user = Auth::user();
        $user->update($request->except('total_price'));

        // Proses checkout
        $code = 'STORE-' . mt_rand(0000,9999);
        $carts = Cart::with(['product','user'])
            ->where('users_id', Auth::user()->id)
            ->get();

        $transaction = Transaction::create([
            'users_id' => Auth::user()->id,
            'insurance_price' => 0,
            'shipping_price' => 0,
            'total_price' => $request->total_price,
            'transaction_status' => 'PENDING',
            'code' => $code
        ]);

        foreach ($carts as $cart) {
            $trx = 'TRX-' . mt_rand(0000,9999);

            TransactionDetail::create([
                'transactions_id' => $transaction->id,
                'products_id' => $cart->product->id,
                'price' => $cart->product->price,
                'shipping_status' => 'PENDING',
                'resi' => '',
                'code' => $trx
            ]);
        }

        // Delete cart data
        Cart::with(['product','user'])
            ->where('users_id', Auth::user()->id)
            ->delete();

        // Konfigurasi midtrans
        Config::$serverKey =
        config('services.midtrans.serverKey');
    }
}

```

```

        Config::$isProduction =
config('services.midtrans.isProduction');
        Config::$isSanitized =
config('services.midtrans.isSanitized');
        Config::$is3ds = config('services.midtrans.is3ds');

        // Buat array untuk dikirim ke midtrans
        $midtrans = array(
            'transaction_details' => array(
                'order_id' => $code,
                'gross_amount' => (int) $request->total_price,
            ),
            // 'item_details' => $items,
            'customer_details' => array(
                'first_name' => Auth::user()->name,
                'email' => Auth::user()->email
            ),
            'enabled_payments' =>
array('gopay', 'bank_transfer'),
            'vtweb' => array()
        );

        try {
            // Ambil halaman payment midtrans
            $paymentUrl = Snap::createTransaction($midtrans)-
>redirect_url;

            // Redirect ke halaman midtrans
            return redirect($paymentUrl);
        }
        catch (Exception $e) {
            echo $e->getMessage();
        }
    }

    public function callback(Request $request)
    {
        // Set konfigurasi midtrans
        Config::$serverKey =
config('services.midtrans.serverKey');
        Config::$isProduction =
config('services.midtrans.isProduction');
        Config::$isSanitized =
config('services.midtrans.isSanitized');
        Config::$is3ds = config('services.midtrans.is3ds');

        // Buat instance midtrans notification

```



```

$notification = new Notification();

// Assign ke variable untuk memudahkan coding
$status = $notification->transaction_status;
$type = $notification->payment_type;
$fraud = $notification->fraud_status;
$order_id = $notification->order_id;

// Cari transaksi berdasarkan ID
$transaction = Transaction::findOrFail($order_id);

// Handle notification status midtrans
if ($status == 'capture') {
    if ($type == 'credit_card'){
        if($fraud == 'challenge'){
            $transaction->status = 'PENDING';
        }
        else {
            $transaction->status = 'SUCCESS';
        }
    }
}
else if ($status == 'settlement'){
    $transaction->status = 'SUCCESS';
}
else if($status == 'pending'){
    $transaction->status = 'PENDING';
}
else if ($status == 'deny') {
    $transaction->status = 'CANCELLED';
}
else if ($status == 'expire') {
    $transaction->status = 'CANCELLED';
}
else if ($status == 'cancel') {
    $transaction->status = 'CANCELLED';
}

// Simpan transaksi
$transaction->save();

// Kirimkan email
if ($transaction)
{
    if($status == 'capture' && $fraud == 'accept' )
    {
        //
    }
}

```

```

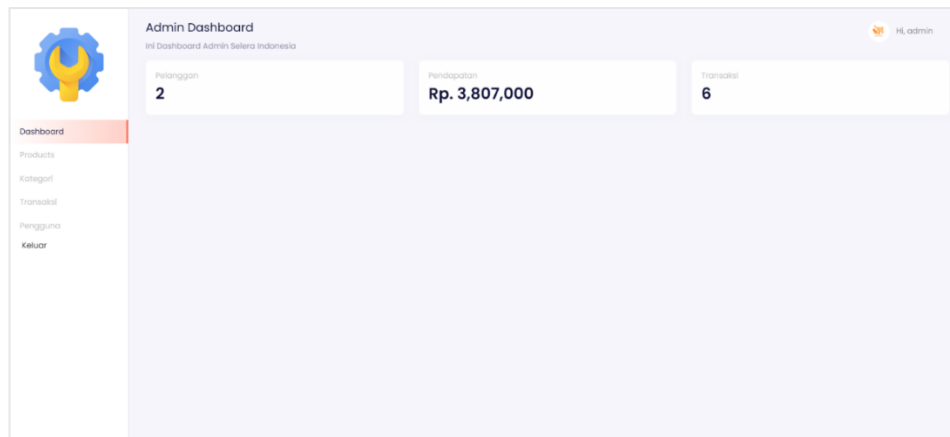
    }
    else if ($status == 'settlement')
    {
        //
    }
    else if ($status == 'success')
    {

    }
    else if($status == 'capture' && $fraud ==
'challenge' )
    {
        return response()->json([
            'meta' => [
                'code' => 200,
                'message' => 'Midtrans Payment
Challenge'
            ]
        ]);
    }
    else
    {
        return response()->json([
            'meta' => [
                'code' => 200,
                'message' => 'Midtrans Payment not
Settlement'
            ]
        ]);
    }

    return response()->json([
        'meta' => [
            'code' => 200,
            'message' => 'Midtrans Notification
Success'
        ]
    ]);
}
}
}

```

8. Tampilan Halaman *Dashboard* Admin



Gambar 4.10 Tampilan Halaman *Dashboard* Admin

Berdasarkan gambar 4.10, halaman *dashboard* admin menunjukkan statistik jumlah pelanggan yang telah terdaftar, pendapatan yang dihasilkan, dan total transaksi yang telah berhasil dilakukan. Dalam konteks ini, hasil dari kode yang terdapat dalam *Controller* adalah sebagai berikut:

```
<?php

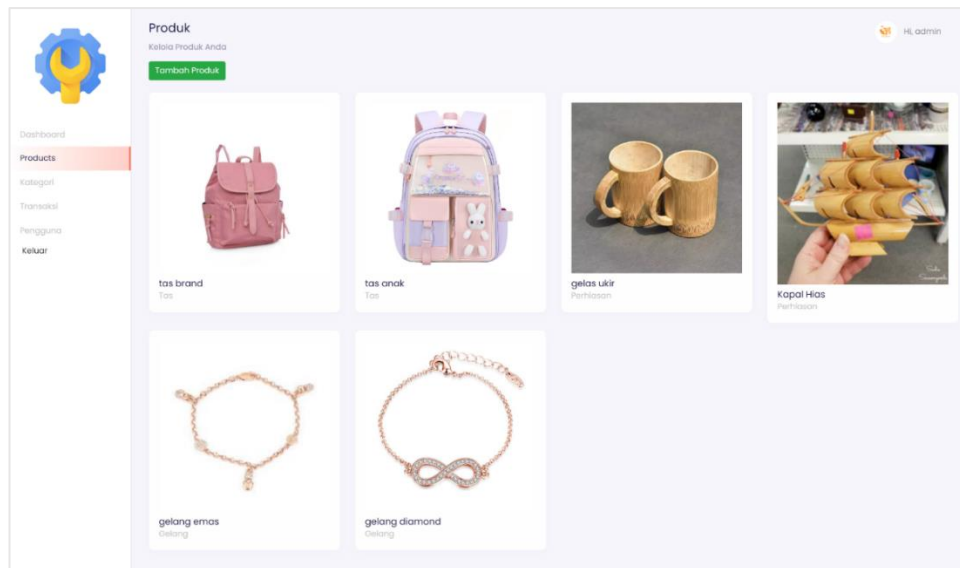
namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\User;
use App\Transaction;

class DashboardController extends Controller
{
    public function index()
    {
        $customer = User::count();
        $revenue = Transaction::sum('total_price');
        $transaction = Transaction::count();

        return view('pages.admin.dashboard', [
            'customer' => $customer,
            'revenue' => $revenue,
            'transaction' => $transaction
        ]);
    }
}
```

9. Tampilan Halaman Produk



Gambar 4.11 Tampilan Halaman Produk

Berdasarkan gambar 4.11, halaman produk ini menampilkan daftar produk beserta opsi untuk menambah produk baru. Berikut adalah hasil implementasi kode yang ada dalam *Controller*:

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use App\Product;
use App\Category;
use App\User;

use App\Http\Requests\Admin\ProductRequest;
use App\ProductGallery;
use Illuminate\Http\Request;
use Illuminate\Support\Str;

class ProductController extends Controller
{

    public function index()
    {
        $products =
        Product::with(['galleries','category'])->get();

        return view('pages.admin.product.products',[
```

```

        'products' => $products
    });
}

public function details(Request $request, $id)
{
    $product =
Product::with(['galleries', 'user', 'category'])-
>findOrFail($id);
    $categories = Category::all();

    return view('pages.admin.product.details', [
        'product' => $product,
        'categories' => $categories
    ]);
}

public function create()
{
    $users = User::all();
    $categories = Category::all();

    return view('pages.admin.product.create', [
        'users' => $users,
        'categories' => $categories
    ]);
}

public function store(ProductRequest $request)
{
    $data = $request->all();

    $data['slug'] = Str::slug($request->name);
    $product = Product::create($data);

    $gallery = [
        'products_id' => $product->id,
        'photos' => $request->file('photo')-
>store('assets/product', 'public')
    ];
    ProductGallery::create($gallery);

    return redirect()->route('product.index');
}

public function show($id)

```

```

    {
    }

    public function edit($id)
    {
        $item = Product::with(['category','user'])->
>findOrFail($id);
        $users = User::all();
        $categories = Category::all();

        return view('pages.admin.product.edit',[
            'item' => $item,
            'users' => $users,
            'categories' => $categories
        ]);
    }

    public function update(ProductRequest $request, $id)
    {
        $data = $request->all();

        $item = Product::findOrFail($id);

        $data['slug'] = Str::slug($request->name);

        $item->update($data);

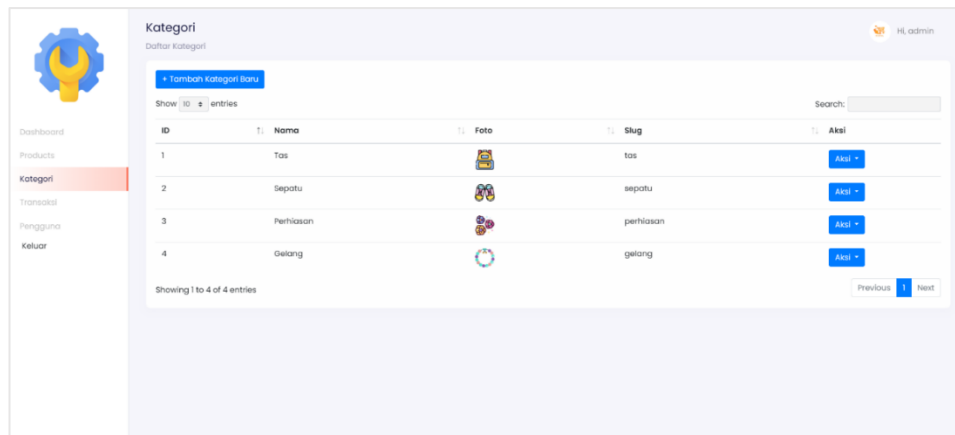
        return redirect()->route('product.index');
    }

    public function destroy($id)
    {
        $item = Product::findOrFail($id);
        $item->delete();

        return redirect()->route('product.index');
    }
}

```

10. Tampilan Halaman Kategori



Gambar 4.12 Tampilan Halaman Kategori

Berdasarkan gambar 4.12, halaman kategori ini menampilkan bagian kategori yang telah diinputkan oleh admin. Sebagai hasilnya, terdapat implementasi kode dalam *Controller* sebagai berikut:

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use App\Category;

use App\Http\Requests\Admin\CategoryRequest;

use Illuminate\Http\Request;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use Yajra\DataTables\Facades\DataTables;

class CategoryController extends Controller
{

    public function index()
    {
        if (request()->ajax()) {
            $query = Category::query();

            return Datatables::of($query)
                ->addColumn('action', function ($item) {
                    return '
                        <div class="btn-group">
```

```

                <div class="dropdown">
                    <button class="btn btn-
primary dropdown-toggle mr-1 mb-1"
                                type="button"
id="action" . $item->id . '"
                                data-
toggle="dropdown"
                                aria-
haspopup="true"
                                aria-
expanded="false">
                        Aksi
                    </button>
                    <div class="dropdown-
menu" aria-labelledby="action" . $item->id . '">
                        <a class="dropdown-
item" href="" . route('category.edit', $item->id) . '">
                            Sunting
                        </a>
                        <form action="" .
route('category.destroy', $item->id) . '" method="POST">
                            ' .
method_field('delete') . csrf_field() . '
                                <button
type="submit" class="dropdown-item text-danger">
                                    Hapus
                                </button>
                            </form>
                        </div>
                    </div>
                </div>;
            })
            ->editColumn('photo', function ($item) {
                return $item->photo ? '' : '';
            })
            ->rawColumns(['action', 'photo'])
            ->make();
        }

        return view('pages.admin.category.index');
    }

    public function create()
    {
        return view('pages.admin.category.create');
    }

```



```

    }

    public function store(CategoryRequest $request)
    {
        $data = $request->all();
        $data['slug'] = Str::slug($request->name);
        $data['photo'] = $request->file('photo')->store('assets/category', 'public');
        Category::create($data);
        return redirect()->route('category.index');
    }

    public function show($id)
    {
    }

    public function edit($id)
    {
        $item = Category::findOrFail($id);

        return view('pages.admin.category.edit', [
            'item' => $item
        ]);
    }

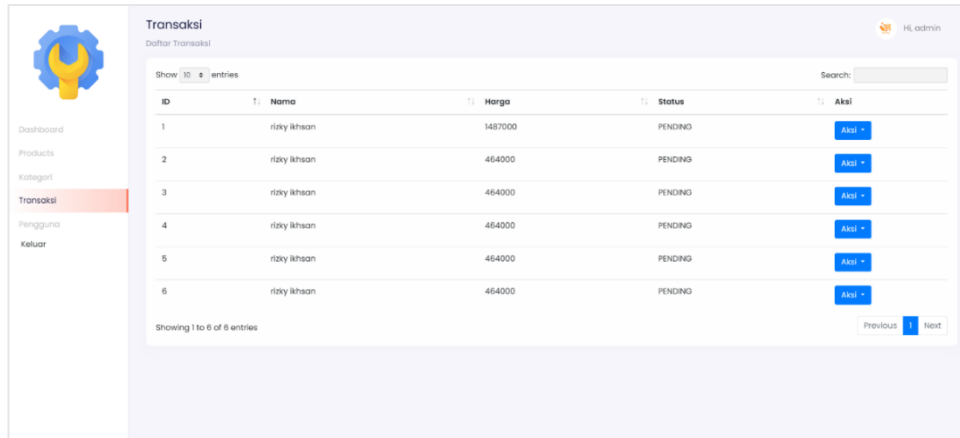
    public function update(CategoryRequest $request, $id)
    {
        $data = $request->all();
        $data['slug'] = Str::slug($request->name);
        $data['photo'] = $request->file('photo')->store('assets/category', 'public');

        $item = Category::findOrFail($id);
        $item->update($data);
        return redirect()->route('category.index');
    }

    public function destroy($id)
    {
        $item = Category::findOrFail($id);
        $item->delete();
        return redirect()->route('category.index');
    }
}

```

11. Tampilan Halaman Transaksi



ID	Nama	Harga	Status	Aksi
1	rizky ikhsan	1487000	PENDING	Aksi
2	rizky ikhsan	464000	PENDING	Aksi
3	rizky ikhsan	464000	PENDING	Aksi
4	rizky ikhsan	464000	PENDING	Aksi
5	rizky ikhsan	464000	PENDING	Aksi
6	rizky ikhsan	464000	PENDING	Aksi

Gambar 4.13 Tampilan Halaman Transaksi

Berdasarkan gambar 4.13, menggambarkan halaman transaksi yang digunakan untuk menampilkan data transaksi yang telah dilakukan oleh pengguna. Oleh karena itu, berikut ini adalah hasil program yang terdapat dalam *Controller* :

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Transaction;
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;

class TransactionController extends Controller
{
    public function index()
    {
        if (request()->ajax()) {
            $query = Transaction::with(['user']);

            return Datatables::of($query)
                ->addColumn('action', function ($item) {
                    return '
                        <div class="btn-group">
                            <div class="dropdown">
                                <button class="btn btn-
primary dropdown-toggle mr-1 mb-1"
                                type="button"
                                id="action' . $item->id . '"
                                data-toggle="dropdown">
                                    <div class="dropdown-menu">
                                        <div class="dropdown-item">
                                            <a href="#">Detail Transaksi</a>
                                        </div>
                                        <div class="dropdown-item">
                                            <a href="#">Hapus Transaksi</a>
                                        </div>
                                    </div>
                            </div>
                        </div>';
                });
        }
    }
}
```

```

toggle="dropdown"
haspopup="true"
expanded="false">
        Aksi
    </button>
    <div class="dropdown-
menu" aria-labelledby="action" . $item->id . '">
        <a class="dropdown-
item" href="' . route('transaction.edit', $item->id) .
'">
            Sunting
        </a>
        <form action="' .
route('transaction.destroy', $item->id) . '"
method="POST">
            ' .
method_field('delete') . csrf_field() . '
            <button
type="submit" class="dropdown-item text-danger">
                Hapus
            </button>
        </form>
    </div>
</div>
</div>';
    })
    ->rawColumns(['action'])
    ->make();
}
$cr = Transaction::get();
return
view('pages.admin.transaction.index', ['cr'=>$cr]);
}

public function create()
{

}

public function store(Request $request)
{

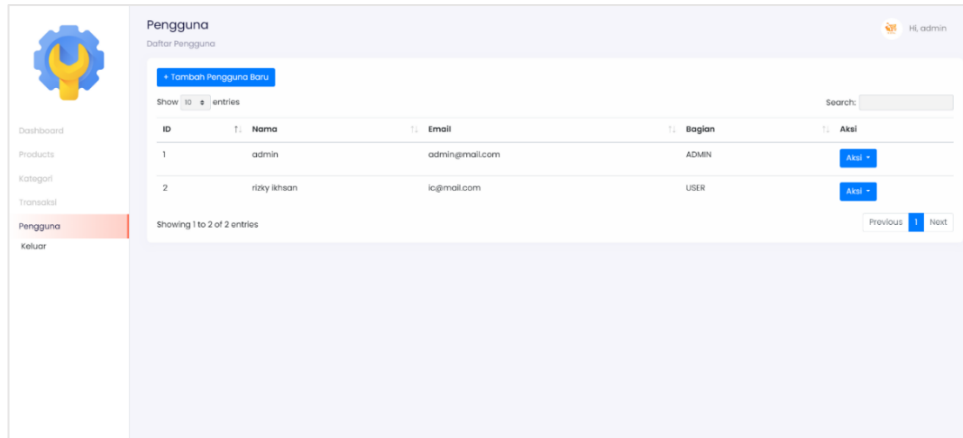
}

public function show($id)

```

```
{  
  
}  
  
public function edit($id)  
{  
    $item = Transaction::with(['user'])->findOrFail($id);  
  
    return view('pages.admin.transaction.edit', [  
        'item' => $item  
    ]);  
}  
  
public function update(Request $request, $id)  
{  
    $data = $request->all();  
    $item = Transaction::findOrFail($id);  
    $item->update($data);  
    return redirect()->route('transaction.index');  
}  
  
public function destroy($id)  
{  
    $item = Transaction::findOrFail($id);  
    $item->delete();  
    return redirect()->route('transaction.index');  
}  
}
```

12. Tampilan Halaman Akun Pengguna



ID	Nama	Email	Bagian	Aksi
1	admin	admin@gmail.com	ADMIN	Aksi
2	rizky ikhsan	ic@gmail.com	USER	Aksi

Gambar 4.14 Tampilan Halaman Akun Pengguna

Berdasarkan Gambar 4.14, data akun dari berbagai pihak, mulai dari pelanggan hingga admin, ditampilkan pada halaman akun pengguna ini. Oleh karena itu, kode yang terdapat dalam *Controller* adalah sebagai berikut:

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use App\User;

use App\Http\Requests\Admin\UserRequest;

use Illuminate\Http\Request;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use Yajra\DataTables\Facades\DataTables;

class UserController extends Controller
{

    public function index()
    {
        if (request()->ajax()) {
            $query = User::query();

            return DataTables::of($query)
                ->addColumn('action', function ($item) {
```

```

        return '
            <div class="btn-group">
                <div class="dropdown">
                    <button class="btn btn-
primary dropdown-toggle mr-1 mb-1"
                                type="button"
id="action" . $item->id . '"
                                data-
toggle="dropdown"
                                aria-
haspopup="true"
                                aria-
expanded="false">
                                    Aksi
                                </button>
                                <div class="dropdown-
menu" aria-labelledby="action" . $item->id . '">
                                    <a class="dropdown-
item" href="" . route('user.edit', $item->id) . '">
                                        Sunting
                                    </a>
                                    <form action="" .
route('user.destroy', $item->id) . '" method="POST">
                                        '
                                        method_field('delete') . csrf_field() . '
                                        <button
type="submit" class="dropdown-item text-danger">
                                            Hapus
                                        </button>
                                    </form>
                                </div>
                            </div>
                        </div>';
    })
    ->rawColumns(['action'])

    ->make();
}

return view('pages.admin.user.index');
}

public function create()
{
    return view('pages.admin.user.create');
}

```

```

    public function store(UserRequest $request)
    {
        $data = $request->all();

        $data['password'] = bcrypt($request->password);

        if ($request->hasFile('photo')) {
            $photoPath = $request->file('photo')->store('assets/user', 'public');
            $data['photo'] = $photoPath;
        }

        User::create($data);

        return redirect()->route('user.index');
    }

    public function show($id)
    {
        //
    }

    public function edit($id)
    {
        $item = User::findOrFail($id);

        return view('pages.admin.user.edit', [
            'item' => $item
        ]);
    }

    public function update(UserRequest $request, $id)
    {
        $data = $request->all();

        $item = User::findOrFail($id);

        if($request->password)
        {
            $data['password'] = bcrypt($request->password);
        }
        else
        {
            unset($data['password']);
        }
    }

```

```

        $item->update($data);

        return redirect()->route('user.index');
    }

    public function destroy($id)
    {
        $item = User::findOrFail($id);
        $item->delete();

        return redirect()->route('user.index');
    }
}

```

4.1.2 Pengujian

Pada tahap ini, dilaksanakan pengujian ketergunaan terhadap *stackholder* Selera Indonesia. Metode pengujian yang digunakan yaitu *System Usability Scale* (SUS) merupakan sebuah kuesioner yang terdiri dari 10 pertanyaan dengan 5 pilihan jawaban, mulai dari sangat tidak setuju hingga sangat setuju. Metode ini memberikan hasil evaluasi yang cepat, sederhana, dan dapat diandalkan untuk mengukur tingkat ketergunaan suatu sistem [15]. SUS dapat digunakan untuk mengevaluasi berbagai macam produk dan layanan seperti perangkat keras, perangkat lunak, perangkat seluler, *website*, dan aplikasi. Manfaat yang didapat menggunakan SUS yaitu :

1. Skala yang sangat mudah untuk diberikan ke responden
2. Dapat digunakan pada ukuran sampel kecil dengan hasil yang andal
3. Secara efektif dapat membedakan sistem yang dapat digunakan dan tidak dapat digunakan.

Pertanyaan yang diberikan pada responden terdiri dari 10 pertanyaan dengan skala 1 sampai 5. Pertanyaan tersebut dapat dilihat pada tabel di bawah dengan keterangan skala sebagai berikut.

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

No	Pertanyaan	Skala				
		1	2	3	4	5
		STS	TS	N	S	SS
1	Saya pikir saya akan sering menggunakan <i>website</i> ini					
2	Saya menemukan fitur yang seharusnya tidak merepotkan					
3	Saya merasa <i>website</i> ini mudah digunakan					
4	Saya sepertinya membutuhkan bantuan seseorang agar bisa lancar menggunakan <i>website</i> ini					
5	Saya pikir fitur yang terdapat pada <i>website</i> ini sudah terintegrasi dengan baik					
6	Saya pikir dalam <i>website</i> ini terdapat banyak hal yang tidak konsisten					
7	Menurut saya, orang lain dapat mempelajari cara menggunakan <i>website</i> ini dengan cepat					
8	Saya merasa <i>website</i> ini sangat sulit untuk digunakan					
9	Saya merasa nyaman menggunakan <i>website</i> ini					
10	Saya perlu belajar banyak hal sebelum saya dapat menggunakan <i>website</i> ini					

Tabel 4.1 Pertanyaan SUS

Terdapat 10 responden yang telah menjawab pertanyaan SUS. Berikut hasil pengujian SUS.

No	Pertanyaan	Skala				
		1	2	3	4	5
		STS	TS	N	S	SS
1	Saya pikir saya akan sering menggunakan <i>website</i> ini					✓
2	Saya menemukan fitur yang seharusnya tidak merepotkan					✓
3	Saya merasa <i>website</i> ini mudah digunakan					✓
4	Saya sepertinya membutuhkan bantuan seseorang agar bisa lancar menggunakan <i>website</i> ini	✓				
5	Saya pikir fitur yang terdapat pada <i>website</i> ini sudah terintegrasi dengan baik					✓
6	Saya pikir dalam <i>website</i> ini terdapat banyak hal yang tidak konsisten		✓			
7	Menurut saya, orang lain dapat mempelajari cara menggunakan <i>website</i> ini dengan cepat					✓
8	Saya merasa <i>website</i> ini sangat sulit untuk digunakan	✓				
9	Saya merasa nyaman menggunakan <i>website</i> ini					✓
10	Saya perlu belajar banyak hal sebelum saya dapat menggunakan <i>website</i> ini		✓			

Pontianak, 26 Agustus 2023

Murni Sabila

Murni Sabila

Gambar 4.15 Pengujian SUS 1

No	Pertanyaan	Skala				
		1	2	3	4	5
		STS	TS	N	S	SS
1	Saya pikir saya akan sering menggunakan <i>website</i> ini				✓	
2	Saya menemukan fitur yang seharusnya tidak merepotkan					✓
3	Saya merasa <i>website</i> ini mudah digunakan					✓
4	Saya sepertinya membutuhkan bantuan seseorang agar bisa lancar menggunakan <i>website</i> ini		✓			
5	Saya pikir fitur yang terdapat pada <i>website</i> ini sudah terintegrasi dengan baik			✓		
6	Saya pikir dalam <i>website</i> ini terdapat banyak hal yang tidak konsisten			✓		
7	Menurut saya, orang lain dapat mempelajari cara menggunakan <i>website</i> ini dengan cepat					✓
8	Saya merasa <i>website</i> ini sangat sulit untuk digunakan		✓			
9	Saya merasa nyaman menggunakan <i>website</i> ini					✓
10	Saya perlu belajar banyak hal sebelum saya dapat menggunakan <i>website</i> ini		✓			

Pontianak, 26 Agustus 2023

Herti
Herti Septiani

Gambar 4.16 Pengujian SUS 2

No	Pertanyaan	Skala				
		1	2	3	4	5
		STS	TS	N	S	SS
1	Saya pikir saya akan sering menggunakan <i>website</i> ini					✓
2	Saya menemukan fitur yang seharusnya tidak merepotkan					✓
3	Saya merasa <i>website</i> ini mudah digunakan					✓
4	Saya sepertinya membutuhkan bantuan seseorang agar bisa lancar menggunakan <i>website</i> ini		✓			
5	Saya pikir fitur yang terdapat pada <i>website</i> ini sudah terintegrasi dengan baik				✓	
6	Saya pikir dalam <i>website</i> ini terdapat banyak hal yang tidak konsisten			✓		
7	Menurut saya, orang lain dapat mempelajari cara menggunakan <i>website</i> ini dengan cepat				✓	
8	Saya merasa <i>website</i> ini sangat sulit untuk digunakan		✓			
9	Saya merasa nyaman menggunakan <i>website</i> ini					✓
10	Saya perlu belajar banyak hal sebelum saya dapat menggunakan <i>website</i> ini	✓				

Pontianak, 26 Agustus 2023



Nabih Iqbal

Gambar 4.17 Pengujian SUS 3

Kemudian jawaban tersebut diolah ke dalam tabel berikut.

Responden	Pertanyaan										Nilai
	1	2	3	4	5	6	7	8	9	10	
1	4	4	4	4	4	3	4	4	4	3	95
2	1	4	4	3	2	2	4	3	4	3	75
3	4	4	4	3	1	2	1	3	4	4	75
Jumlah											240
Skor SUS											80

Tabel 4.2 Hasil SUS

<i>Acceptability Range</i>	Rentang Skor
<i>Not Acceptable</i>	0-50
<i>Marginal</i>	50-70
<i>Acceptable</i>	70-100

Tabel 4.3 SUS Acceptability Score

Hasil Pengujian SUS pada tabel 4.2 di atas mendapatkan hasil *acceptable* karena memiliki skor di atas 70 sehingga dapat diterima dan siap digunakan pengguna.

Dokumentasi mengenai pengujian penggunaan sistem yang telah dilakukan kepada pihak Selera Indonesia melalui wawancara telah disiapkan sebagai berikut:



Gambar 4.18 Pengujian Oleh *Stakeholder*



Gambar 4.19 Pengujian Pada Bagian Midtrans dan Raja Ongkir

4.2. Implementasi Akhir (*Cutover*)

Pada tahap ini, memastikan bahwa seluruh data, alur, tampilan, proses, dan fungsi beroperasi dengan baik menjadi prioritas, sambil berupaya semaksimal mungkin untuk mencegah terjadinya kesalahan atau kekurangan. Selain itu, tahap ini juga melibatkan pengecekan untuk meningkatkan stabilitas *website*, dengan pemeriksaan yang cermat terhadap antarmuka yang telah dibuat. Kolaborasi dengan beberapa individu dilakukan dalam keputusan menjalankan langkah ini, dengan tujuan untuk melakukan pemeriksaan menyeluruh terhadap fungsionalitas *website* guna mencegah kesalahan serius.

4.3. Hasil Pengujian Seluruh Sistem

Setelah penulis melakukan pengujian terhadap beberapa *stakeholder Selera Indonesia* melalui SUS yang dilakukan secara bertemu langsung, dibawah ini adalah uraian dari daftar fitur beserta hasil pengujian keseluruhan sistem yang ada di Selera Indonesia:

1) Fitur pendataan produk di *dashboard* admin

Hasil : tester dapat memahami alur kerja serta informasi data yang disajikan di halaman tersebut dengan baik.

2) Fitur Ongkos Kirim

Hasil : tester dapat memahami informasi data ongkir yang disajikan di halaman tersebut dengan baik.

3) Fitur Pembayaran

Hasil : tester dapat memahami alur kerja serta informasi data yang disajikan di halaman tersebut dengan baik.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil uraian dari penjelasan sebelumnya, penulis mengambil beberapa kesimpulan antara lain:

1. Berdasarkan hasil analisa, navigasi-navigasi pada *dashboard customer* dan admin cukup mudah dipahami dan informasi yang disajikan juga sudah sangat membantu.
2. Hasil *usability testing* yang dijalankan cukup berhasil dari *stakeholder* secara langsung.
3. *Website* Selera Indonesia ini berguna sebagai penunjang jual beli produk antara admin sebagai penjual dan *customer* sebagai pembeli dengan berbagai fitur dan navigasi yang terstruktur sehingga juga dapat memudahkan admin untuk memantau aktivitas transaksi di satu tempat yang sama secara *online*.
4. Fitur yang di implementasikan sekarang ini merupakan fitur dasar yang dapat menjadi penunjang proses jual beli di UMKM Selera Indonesia dan hal tersebut tidak akan menutup kemungkinan untuk melakukan implementasi fitur-fitur lainnya nantinya.
5. Proses pengembangan *website* Selera Indonesia ini dilakukan secara bertahap dimulai dari riset dan pembuatan *wireframe* yang mana hal ini sangat berguna jika sewaktu-waktu akan ada perubahan struktur, maka dapat dimulai dari *wireframe*.

5.2. Saran

Berdasarkan *usability testing* yang telah dilakukan pada *website* ini, terdapat beberapa kekurangan yang dapat membantu dalam pengembangan sistem antara lain :

1. Belum adanya notifikasi ke *WA Gateway* untuk notifikasi batas waktu pembayaran
2. Pada integrasi fitur ongkir terdapat perhitungan ongkos kirim yang mahal sehingga calon *customer* kurang berminat membeli produk.
3. Belum adanya pengaturan stok produk dari *website* Selera Indonesia dikarenakan dari penjual produk bakal diadakan stok selalu ada.

DAFTAR PUSTAKA

- [1] W. F. Sahri, “SISTEM INFORMASI PON-PES HUDATUL MUNA 2 PONOROGO BERBASIS WEB YANG DILENGKAPI DENGAN FORMULIR PENDAFTARAN SANTRI ON-LINE,” Ponorogo, 2013.
- [2] A. D. Putri, B. Permatasari dan E. Suwarni, “STRATEGI DESAIN KEMASAN SEBAGAI UPAYA PENINGKATAN DAYA JUAL PRODUK UMKM KELURAHAN LABUHAN DALAM BANDARLAMPUNG,” *Social Sciences and Technology for Community Service (JSSTCS)*, vol. 4, no. 1, pp. 119-123, 2023.
- [3] S. A. Triana, *Perancangan Website Sweda Custumring Sebagai Upaya Membangun Brand Image*, Yogyakarta: ISI Yogyakarta, 2019.
- [4] R. Abdurrofi, *PEMBUATAN SISTEM INFORMASI E-COMMERCE BERBASIS WEB PADA UMKM BINAAN ASPAMI*, Bandung: UNIVERSITAS WIDYATAMA, 2017.
- [5] C. YUNITA, *RANCANG BANGUN WEBSITE E-COMMERCE KUSTOMISASI PEMESANAN PRODUK HANDICRAFT (STUDI KASUS : UMKM NENA NAMO)*, Surabaya: Institut Teknologi Sepuluh Nopember, 2018.
- [6] P. Magdalena dan A. R. Putera, “Rancang Bangun Sistem Informasi Pemasaran Batik Sembogem Pada Umkm Desa Bogem Magetan,” *Seminar Nasional Teknologi Informasi dan Komunikasi-2021*, vol. 4, no. 1, pp. 659-661, 2021.
- [7] P. Magdalena dan A. R. Putera, “Rancang Bangun Sistem Informasi Pemasaran Batik Sembogem Pada Umkm Desa Bogem Magetan,” *Optimalisasi Digital Marketing Menyongsong Marketing 5.0:Technology For Humanity*, vol. 4, no. 1, pp. 659-667, 2021.
- [8] L. Indrayani, F. Awopi dan S. Zabir, “SISTEM INFORMASI PEMASARAN USAHA KECIL MENENGAH (UKM) BERBASIS WEBSITE,” *DINAMIS*, vol. 17, no. 1, pp. 111-116, 2020.

- [9] F. A. , “Pengertian Website, Manfaat, dan Jenis-Jenisnya Lengkap!,” Hostinger, 23 Desember 2022. [Online]. Available: <https://www.hostinger.co.id/tutorial/website-adalah>. [Diakses 2023 Maret 13].
- [10] A. Fauzi, “7 Pengertian Website Menurut Ahli, Lengkap Jenis & Fungsinya,” CNBC Indonesia, 18 Juni 2022. [Online]. Available: <https://www.cnbcindonesia.com/tech/20220618152119-37-348229/7-pengertian-website-menurut-ahli-lengkap-jenis-fungsinya>. [Diakses 13 Maret 2023].
- [11] A. Rahman, Pengembangan Sistem Informasi Penjualan Berbasis Web Pada Toko Buku Online, Depok: Universitas Nurul Fikri, 2017.
- [12] J. Yonata, “Mengenal DBMS: Pengertian, Jenis, dan Fungsinya,” dewa web, 12 Agustus 2021. [Online]. Available: <https://www.dewaweb.com/blog/apa-itu-DBMS/>. [Diakses 13 Maret 2023].
- [13] F. W., “Apa Itu DBMS? Pengertian, Jenis, dan Contoh DBMS,” Hostinger, 18 Januari 2023. [Online]. Available: <https://www.hostinger.co.id/tutorial/DBMS-adalah>. [Diakses 13 Maret 2023].
- [14] H. Aliya, “DBMS: Definisi, Fungsi, Jenis, Kelebihan dan Kekurangannya,” glints, 16 September 2022. [Online]. Available: <https://glints.com/id/lowongan/dmbs-database-management-system-adalah/>. [Diakses 13 Maret 2023].
- [15] usability.gov, “System Usability Scale (SUS),” usability.gov, [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Diakses 11 September 2023].