

Pengembangan Sistem E-Commerce Elemen Bangunan Menggunakan Next.js dan Midtrans pada UD. Wahyu Jati

Zidan Bayu Mahendra^{*1}, Lestari Retnawati²

^{1,2} Program Studi Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

Email: ¹zbayumahendra@gmail.com, ²lestari@uwks.ac.id

^{*}Penuli Koresponden

Abstrak

Perkembangan teknologi informasi mendorong pelaku Usaha Mikro, Kecil, dan Menengah (UMKM) untuk melakukan digitalisasi proses bisnis guna meningkatkan daya saing dan efektivitas operasional. UD. Wahyu Jati, sebuah UMKM di Surabaya yang bergerak dalam penjualan elemen bangunan yaitu kusen, pintu, jendela dan lain-lain yang saat ini masih menjalankan transaksi secara konvensional mulai dari proses pemesanan, pencatatan stok, hingga pembayaran. Kondisi ini menimbulkan berbagai kendala seperti keterbatasan jangkauan pasar, ketidakefisienan pencatatan, serta potensi kesalahan dalam proses transaksi. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem informasi *e-commerce* berbasis web yang mampu mendigitalisasi proses penjualan UD. Wahyu Jati serta mengintegrasikan sistem pembayaran otomatis menggunakan *Midtrans*.

Pengembangan sistem dilakukan menggunakan metode *waterfall* melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian. Teknologi yang digunakan meliputi *Next.js* sebagai *backend* dengan dukungan *Server-Side Rendering* untuk optimasi kecepatan dan *SEO*, serta *HTML*, *CSS*, dan *JavaScript* pada sisi *frontend* untuk membangun antarmuka yang responsif. Integrasi *Midtrans* diimplementasikan untuk menyediakan berbagai metode pembayaran seperti *transfer bank*, *virtual account*, *QRIS*, dan *e-wallet* sehingga proses transaksi dapat berlangsung lebih cepat, aman, dan terotomatisasi. Sistem yang dibangun mencakup fitur katalog produk, keranjang belanja, *checkout*, pembayaran digital, serta manajemen produk pada sisi Admin.

Kata kunci: *E-commerce*, UMKM, *Midtrans*, *Next.js*, Sistem Informasi, *Payment gateway*.

Abstract

Advances in information technology are encouraging Micro, Small, and Medium Enterprises (MSMEs) to digitize their business processes to improve competitiveness and operational effectiveness. UD. Wahyu Jati, an MSME in Surabaya that sells building components such as frames, doors, windows, and other items, currently still conducts transactions conventionally, from ordering to stock recording to payment. This situation creates various obstacles such as limited market reach, inefficient recording, and the potential for errors in the transaction process. Therefore, this research aims to design and build a web-based *e-commerce* information system capable of digitizing UD. Wahyu Jati's sales process and integrating an automated payment system using *Midtrans*.

The system was developed using the *waterfall* method, encompassing the stages of requirements analysis, design, implementation, and testing. The technologies used include *Next.js* as the backend with support for *Server-Side Rendering* for speed and *SEO* optimization, and *HTML*, *CSS*, and *JavaScript* on the frontend to build a responsive interface. *Midtrans* integration was implemented to provide various payment methods, such as bank transfers, virtual accounts, *QRIS*, and *e-wallets*, allowing for faster, more secure, and automated transactions. The system includes a Product catalog, shopping cart, *checkout*, digital payments, and Product management features for the admin side..

Keywords: *Information System*, *E-commerce*, Micro Small and Medium Enterprises, *Midtrans*, *Next.js*, *Payment gateway*.

I. PENDAHULUAN

Perkembangan teknologi informasi pada era digital saat ini memberikan dampak signifikan terhadap berbagai sektor, termasuk sektor perdagangan. Perubahan perilaku konsumen yang semakin mengandalkan internet dalam mencari informasi produk, melakukan pemesanan, hingga menyelesaikan transaksi, membuat banyak pelaku usaha dituntut untuk beradaptasi [1]. Digitalisasi menjadi langkah strategis yang mampu meningkatkan daya saing, efisiensi operasional, serta memperluas jangkauan pasar. Pada konteks usaha mikro, kecil, dan menengah (UMKM), penerapan teknologi digital seperti *e-commerce* tidak hanya meningkatkan efektivitas proses bisnis, tetapi juga menjadi kunci keberlangsungan usaha di tengah persaingan yang semakin ketat.[2]

UD. Wahyu Jati merupakan salah satu UMKM yang berlokasi di Surabaya, bergerak di bidang penjualan kusen kayu, pintu, jendela, dan produk berbahan dasar kayu lainnya. Seluruh aktivitas penjualan dan pelayanan pelanggan masih dilakukan secara konvensional di wilayah Surabaya, sehingga proses bisnisnya memiliki keterbatasan dalam menjangkau konsumen di luar area tersebut.

Penerapan *e-commerce* menjadi solusi strategis untuk mengatasi permasalahan tersebut. Dengan adanya platform *e-commerce*, UD. Wahyu Jati Surabaya dapat menampilkan katalog produk secara online, menyediakan informasi harga dan varian produk, serta memungkinkan pelanggan melakukan pemesanan tanpa harus datang langsung ke tempat usaha[3]. Integrasi dengan *payment gateway* seperti *Midtrans* membuat proses pembayaran menjadi lebih praktis, aman, dan cepat. *Midtrans* menyediakan berbagai metode pembayaran seperti transfer bank, *virtual account*, *e-wallet*, dan kartu kredit, yang sangat relevan bagi konsumen yang menginginkan fleksibilitas dalam melakukan transaksi[4].

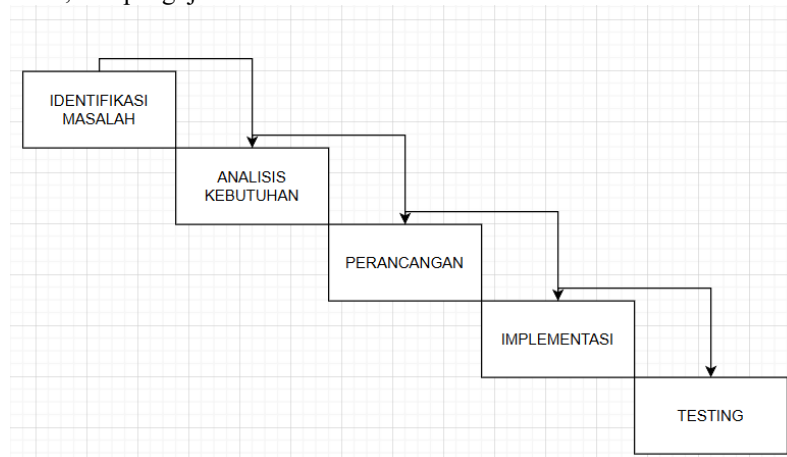
Dalam proses pengembangannya, sistem *e-commerce* ini dibangun menggunakan teknologi modern berbasis *Next.js* sebagai *backend* yang mendukung konsep *Server-Side Rendering (SSR)*. Teknologi tersebut dipilih karena mampu meningkatkan performa akses website, optimasi mesin pencari (*SEO*), serta memberikan pengalaman pengguna yang lebih baik. Pada sisi frontend, sistem menggunakan *HTML*, *CSS*, dan *JavaScript* untuk membangun tampilan antarmuka yang responsif, interaktif, dan mudah digunakan oleh pengguna[5]. Pemanfaatan teknologi tersebut diharapkan mampu menghasilkan sistem yang modern, stabil, serta mudah dikembangkan pada masa mendatang sesuai kebutuhan bisnis UD. Wahyu Jati.

Berdasarkan beberapa penelitian terdahulu, implementasi *e-commerce* yang terintegrasi dengan *payment gateway* terbukti mampu meningkatkan efisiensi operasional dan performa bisnis UMKM secara signifikan. Integrasi *Midtrans* memungkinkan proses transaksi berlangsung secara otomatis, cepat, dan aman sehingga mampu meningkatkan kenyamanan pelanggan dalam melakukan pembayaran. Selain itu, sistem pembayaran digital juga terbukti memberikan pengaruh positif terhadap peningkatan kinerja keuangan UMKM karena proses transaksi menjadi lebih praktis dan profesional.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem informasi *e-commerce* elemen bangunan berbasis *web* pada UD. Wahyu Jati dengan mengintegrasikan *payment gateway Midtrans* sebagai sistem pembayaran digital. Sistem yang dikembangkan diharapkan mampu membantu proses digitalisasi bisnis, meningkatkan kualitas pelayanan pelanggan, memperluas jangkauan pemasaran, serta meningkatkan efektivitas dan efisiensi operasional usaha secara menyeluruh. Selain itu, penelitian ini juga diharapkan dapat menjadi referensi dan contoh implementasi digitalisasi bagi UMKM lain yang masih menjalankan proses bisnis secara manual.

II. METODE PENELITIAN

Pada penelitian ini mengadopsi metode pengembangan perangkat lunak yang bersifat sistematis dan terstruktur yaitu metode *Waterfall*[6], yang terdiri dari tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian.



Gambar 1. Metode *Waterfall*

2.1. Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi permasalahan serta kebutuhan sistem yang akan dikembangkan. Pengumpulan data dilakukan melalui observasi, wawancara, dan studi literatur. Hasil analisis menunjukkan bahwa sistem yang berjalan masih menggunakan pencatatan manual, sehingga diperlukan sistem terkomputerisasi untuk meningkatkan efisiensi dan akurasi data.

Kebutuhan sistem dibagi menjadi dua, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi pengelolaan produk, parameter kustomisasi, pesanan serta pembayaran. Sementara itu,

kebutuhan non-fungsional mencakup aspek keamanan sistem, kemudahan penggunaan, serta kompatibilitas perangkat.

Tabel 1. Pengguna Sistem

Pengguna	Aktivitas	Kebutuhan
Admin	<i>Login</i> , kelola parameter kustomisasi, kelola pesanan, kelola produk,	Input, edit, hapus data
User	<i>Login</i> , lihat produk, kustomisasi, pesan, transaksi	Akses informasi

2.2. Perancangan Sistem

Tahap perancangan dilakukan untuk menggambarkan arsitektur dan alur kerja sistem yang akan dibangun. Perancangan sistem meliputi *Class Diagram* dan *Use case diagram*. Selain itu, dilakukan juga perancangan antarmuka pengguna (*user interface*) untuk mempermudah interaksi antara pengguna dengan sistem.

2.3. Implementasi Sistem

Tahap implementasi merupakan proses pengkodean berdasarkan hasil perancangan sistem. Sistem dikembangkan berbasis *web* menggunakan bahasa pemrograman *frontend* nya *html*, *css* dan *javascript* dan *backend*nya *Next.js* serta *MySQL* sebagai basis data. Sistem ini dirancang untuk dapat diakses oleh dua jenis pengguna, yaitu admin *website* dan *user*(pengguna).

Dalam sistem ini juga diterapkan kustomisasi agar *user* (pengguna) dapat meng kustom produk nya sendiri sesuai selera

2.4. Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Metode pengujian yang digunakan adalah *Black Box Testing*, yang berfokus pada pengujian fungsi sistem tanpa melihat struktur internal program.

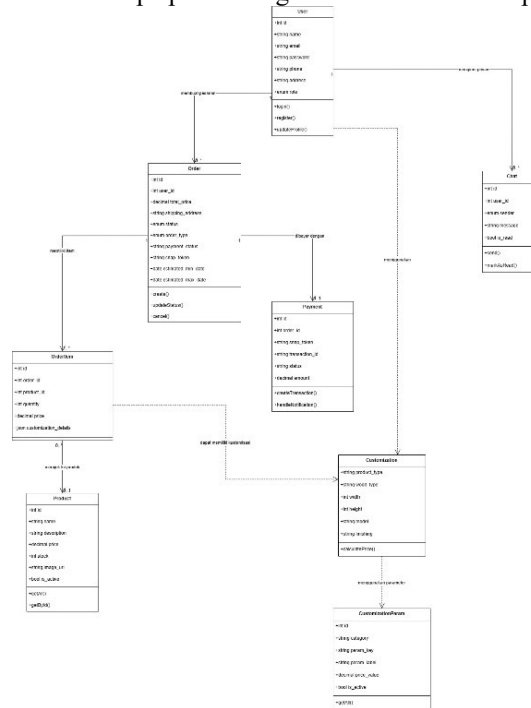
Pengujian dilakukan pada seluruh fitur utama sistem, seperti proses *login*, pengelolaan produk, pengelolaan transaksi, serta parameter kustomisasi. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik sesuai dengan kebutuhan pengguna dan tidak ditemukan kesalahan fungsional.

III. HASIL DAN PEMBAHASAN

3.1. Class Diagram User

Pada Gambar 2. Bagian pengguna, sistem menggunakan *class User* untuk merepresentasikan pelanggan yang menggunakan aplikasi *e-commerce*. Di dalam *class* ini terdapat atribut *id*, *name*, *email*, *password*, *phone*, *address*, dan *role*. Atribut-atribut tersebut digunakan untuk menyimpan identitas pelanggan di dalam basis data. *Method register()* digunakan untuk proses registrasi akun baru, *method login()* digunakan untuk autentikasi pengguna, sedangkan *method updateProfile()* digunakan untuk memperbarui data profil pengguna. *Class User* memiliki relasi dengan *class Order*, yang menunjukkan bahwa satu pengguna dapat membuat banyak pesanan. Relasi ini merupakan implementasi hubungan *one-to-many*. Ketika pelanggan melakukan pemesanan, data transaksi akan disimpan pada *class Order* yang berisi informasi total harga, alamat pengiriman, status pesanan, serta status pembayaran. Dalam setiap transaksi, sistem menggunakan *class OrderItem* untuk menyimpan detail produk yang dipesan oleh pelanggan. *Class* ini memiliki atribut *id*, *order_id*, *Product_id*, *quantity*, *price*, dan *customization_details*. Atribut *customization_details* menggunakan tipe data *json*, sehingga sistem dapat menyimpan detail konfigurasi produk secara fleksibel tanpa harus menambahkan banyak kolom baru pada tabel basis data. Pendekatan ini sangat sesuai untuk sistem berbasis kustomisasi karena setiap produk dapat memiliki konfigurasi yang berbeda-beda. *Class OrderItem* memiliki relasi dengan *class Product*, di mana setiap item pesanan merujuk pada satu produk tertentu. Relasi tersebut menunjukkan implementasi *many-to-one relationship* karena banyak item pesanan dapat mengacu pada produk yang sama. Dalam proses pembayaran, sistem menggunakan *class Payment* yang terintegrasi dengan *payment gateway Midtrans*. *Class* ini memiliki atribut *id*, *order_id*, *snap_token*, *transaction_id*, *status*, dan *amount*. Atribut *snap_token* digunakan sebagai token transaksi pembayaran dari *Midtrans* yang digunakan pelanggan ketika melakukan pembayaran. *Method createTransaction()* digunakan untuk membuat transaksi pembayaran secara otomatis, sedangkan *method handleNotification()* digunakan untuk menerima notifikasi pembayaran dari sistem *Midtrans* terkait status transaksi pelanggan. Fitur utama dalam sistem ini adalah fitur kustomisasi produk yang direpresentasikan melalui *class Customization*. *Class* ini memungkinkan pelanggan menentukan spesifikasi produk sesuai kebutuhan. Atribut yang terdapat di dalamnya meliputi

Product_type, *wood_type*, *width*, *height*, *model*, dan *finishing*. Data tersebut digunakan untuk menentukan karakteristik produk yang akan diproduksi oleh pihak UD. Wahyu Jati sesuai permintaan pelanggan. Selain atribut, *class Customization* juga memiliki *method calculatePrice()* yang digunakan untuk menghitung harga akhir produk berdasarkan kombinasi parameter yang dipilih pengguna. Perhitungan harga dilakukan secara dinamis dengan memanfaatkan data parameter yang berasal dari *class CustomizationParam*. *Class Customization* memiliki hubungan *dependency* dengan *class CustomizationParam*. Relasi ini menunjukkan bahwa proses kustomisasi produk menggunakan parameter yang telah didefinisikan sebelumnya oleh administrator. Dengan pendekatan tersebut, sistem menjadi lebih fleksibel karena administrator dapat memperbarui parameter kustomisasi tanpa perlu mengubah struktur utama aplikasi maupun basis data.

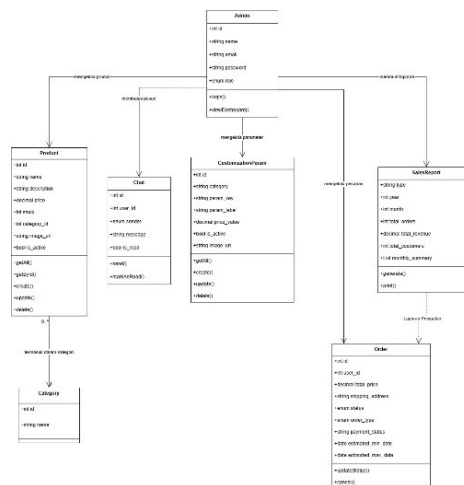


Gambar 2. Class Diagram User

3.2. Class Diagram Admin

Pada bagian admin yang ada pada Gambar 3. *Class* utama yang digunakan adalah *class Admin*. *Class* ini merepresentasikan administrator sistem yang memiliki hak akses penuh terhadap pengelolaan data dan aktivitas operasional aplikasi. Di dalam *class Admin* terdapat atribut *id*, *name*, *email*, *password*, dan *role*. Atribut *id* berfungsi sebagai primary key yang digunakan untuk membedakan setiap data administrator secara unik di dalam basis data. Atribut *email* dan *password* digunakan dalam proses autentikasi ketika admin melakukan *login* ke dalam sistem. Sementara itu, atribut *role* bertipe *enum* yang digunakan untuk menentukan level hak akses administrator. Selain atribut, *class Admin* memiliki *method login()* yang digunakan untuk melakukan autentikasi ke sistem serta *method viewDashboard()* yang berfungsi menampilkan halaman dashboard berisi ringkasan aktivitas sistem. Melalui dashboard tersebut, administrator dapat memantau data produk, transaksi, laporan penjualan, serta aktivitas pelanggan. *Class Admin* memiliki relasi dengan *class Product* karena administrator bertugas mengelola seluruh data produk yang tersedia dalam sistem. *Class Product* digunakan untuk menyimpan informasi mengenai produk elemen bangunan yang dijual oleh UD. Wahyu Jati. Di dalam *class* ini terdapat atribut *id*, *name*, *description*, *price*, *stock*, *category_id*, *image_url*, dan *is_active*. Atribut *category_id* berfungsi sebagai foreign key yang menghubungkan produk dengan *class Category*. Dengan adanya relasi tersebut, setiap produk dapat dikelompokkan ke dalam kategori tertentu sehingga proses pengelolaan data menjadi lebih terstruktur. *Class Product* juga memiliki beberapa *method* seperti *getAll()*, *getById()*, *create()*, *update()*, dan *delete()*. *Method-method* tersebut mendukung proses *CRUD* (*Create*, *Read*, *Update*, *Delete*) terhadap data produk. Administrator dapat menambahkan produk baru, memperbarui informasi produk, menampilkan data produk, maupun menghapus produk yang sudah tidak digunakan lagi dalam sistem. Relasi antara *class Product* dan *Category* menunjukkan hubungan *one-to-many*, di mana satu kategori dapat memiliki banyak produk, sedangkan satu produk hanya berada pada satu kategori tertentu. *Class Category* sendiri memiliki

atribut *id* dan *name*. Walaupun sederhana, *class* ini memiliki peran penting dalam menjaga normalisasi basis data agar data kategori tidak mengalami redundansi. Selain mengelola produk, administrator juga dapat berkomunikasi dengan pelanggan melalui *class* Chat. *Class* ini digunakan sebagai media komunikasi antara admin dan user terkait konsultasi produk, konfirmasi pesanan, maupun layanan pelanggan lainnya. Atribut yang terdapat pada *class* Chat meliputi *id*, *user_id*, *sender*, *message*, dan *is_read*. Atribut *sender* bertipe *enum* yang digunakan untuk membedakan apakah pesan dikirim oleh admin atau user. Sedangkan atribut *is_read* digunakan untuk menandai status pesan apakah sudah dibaca atau belum. *Method* *send()* digunakan untuk mengirim pesan dan *method* *markAsRead()* digunakan untuk mengubah status pesan menjadi telah dibaca. Sistem juga menyediakan fitur kustomisasi produk yang dikelola melalui *class* *CustomizationParam*. *Class* ini digunakan untuk menyimpan parameter kustomisasi seperti jenis kayu, model desain, *finishing*, maupun ukuran tertentu. Di dalam *class* ini terdapat atribut *id*, *category*, *param_key*, *param_label*, *price_value*, *is_active*, dan *image_url*. Atribut *price_value* digunakan untuk menentukan nilai tambahan harga berdasarkan parameter yang dipilih pelanggan. Dengan demikian, sistem dapat menghitung harga produk secara dinamis sesuai konfigurasi yang dipilih pengguna. *Class* *CustomizationParam* memiliki *method* *getAll()*, *create()*, *update()*, dan *delete()* yang digunakan untuk mendukung proses pengelolaan parameter kustomisasi oleh administrator. Pendekatan ini menunjukkan bahwa sistem dirancang dengan fleksibilitas tinggi karena administrator dapat menambah atau memperbarui parameter tanpa perlu mengubah struktur utama sistem maupun basis data. Dalam proses transaksi, administrator juga mengelola *class* *Order*. *Class* ini digunakan untuk merepresentasikan data pemesanan pelanggan. Atribut yang terdapat pada *class* *Order* meliputi *id*, *user_id*, *total price*, *shipping address*, *status*, *order type*, *payment status*, *estimated min date*, dan *estimated max date*. Atribut *status* digunakan untuk menunjukkan kondisi pesanan seperti diproses, dikirim, atau selesai. Sedangkan atribut *payment_status* menunjukkan status pembayaran pelanggan. Informasi estimasi pengiriman disimpan pada atribut *estimated_min_date* dan *estimated_max_date*. *Class* *Order* memiliki *method* *updateStatus()* yang digunakan administrator untuk memperbarui status pesanan sesuai perkembangan proses produksi atau pengiriman. Selain itu, *method* *cancel()* digunakan untuk membatalkan transaksi apabila diperlukan. Untuk mendukung proses analisis dan monitoring penjualan, sistem menyediakan *class* *SalesReport*. *Class* ini digunakan untuk menghasilkan laporan penjualan berdasarkan periode tertentu. Atribut pada *class* ini meliputi *type*, *year*, *month*, *total orders*, *total revenue*, *total customers*, dan *monthly summary*. Data tersebut digunakan untuk menampilkan statistik penjualan yang dapat membantu pihak manajemen dalam pengambilan keputusan bisnis. *Method* *generate()* digunakan untuk membuat laporan secara otomatis, sedangkan *method* *print()* digunakan untuk mencetak laporan penjualan.

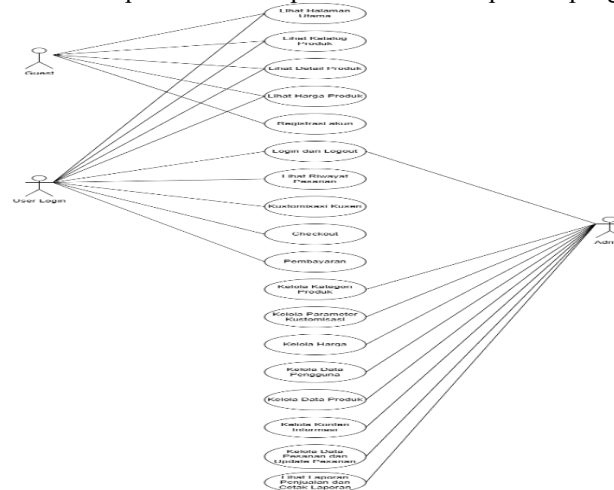


Gambar 3. Class Diagram Admin

3.3. Use Case Diagram

Aktor sistem secara keseluruhan terdiri dari pengguna (*user*) dan administrator (*admin*) yang saling terintegrasi dalam satu alur layanan. Pengguna (*user*) merupakan satu kesatuan aktor yang dibedakan berdasarkan status autentikasi, yaitu *guest user* dan *user login*. *Guest user* memiliki hak akses terbatas pada fitur informatif, seperti melihat halaman utama, katalog produk, detail produk, dan harga produk, serta melakukan registrasi akun. Setelah melakukan login, status pengguna meningkat menjadi

user login dengan hak akses penuh terhadap fitur transaksional, seperti kustomisasi kusen, *checkout*, pembayaran, serta melihat riwayat pesanan. Sementara itu, *admin* berperan sebagai aktor pengelola sistem yang bertanggung jawab terhadap keberlangsungan dan kualitas layanan. *Admin* memiliki kewenangan untuk mengelola kategori produk, parameter kustomisasi, harga, data pengguna, dan data produk, serta mengelola konten informasi. Selain itu, admin mengelola data pesanan dan pembaruan status pesanan untuk memastikan proses transaksi berjalan dengan baik. *Admin* juga melakukan pemantauan kinerja sistem melalui fitur laporan penjualan dan pencetakan laporan sebagai dasar evaluasi dan pengambilan keputusan. Integrasi antara *user* dan *admin* ini membentuk satu kesatuan sistem *e-commerce* kusen yang aman, terstruktur, dan berorientasi pada efisiensi operasional serta kepuasan pengguna.



Gambar 4. Use Case Diagram

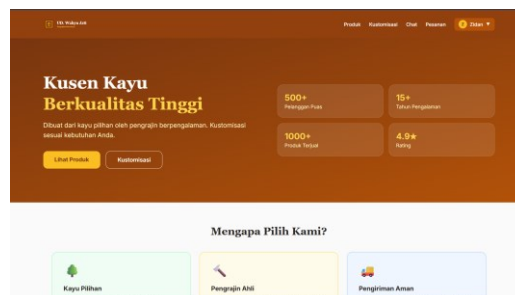
3.4. Halaman Login

Halaman *login* digunakan sebagai pintu masuk sistem bagi *user* dan admin dengan menggunakan *e-mail* dan *password*.

Gambar 5. Halaman Login

3.5. Halaman Dashboard User

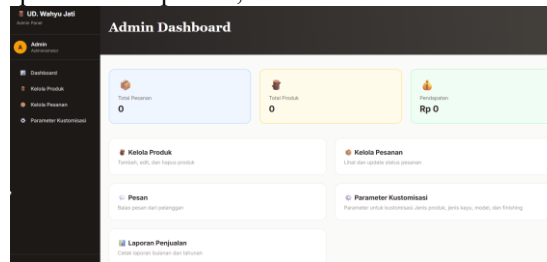
Dashboard menampilkan informasi utama sistem yang dapat diakses oleh *user* untuk kebutuhannya dalam web.



Gambar 6. Halaman Dashboard User

3.6. Halaman *Dashboard Admin*

Dashboard menampilkan informasi utama sistem yang dapat diakses oleh admin untuk mengelola web yang digunakan *user* seperti tambah produk, edit dan lain-lain



Gambar 7. Halaman *Dashboard Admin*

3.7. Halaman Registrasi

Halaman registrasi *user* menampilkan formulir pendaftaran akun baru sebagaimana ditunjukkan pada Gambar 9. *User* diwajibkan mengisi data diri meliputi nama lengkap, *email*, *password*, no HP, dan Alamat.

Gambar 8. Halaman Registrasi

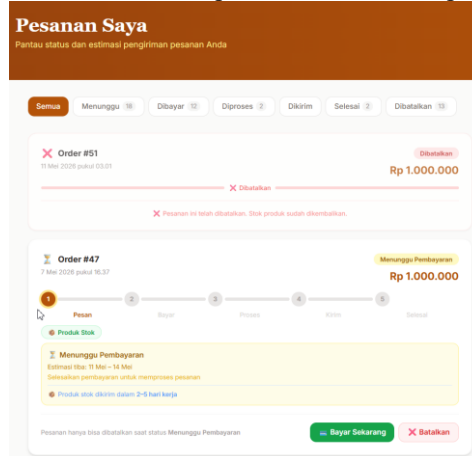
3.8. Halaman Kustomisasi

Halaman kustomisasi untuk pemesanan produk kustomisasi yang diinginkan oleh *user*. Didalam halaman ini sudah ada perhitungan otomatis dan pembayaran yang telah di integrasi oleh *Midtrans*.

Gambar 9. Halaman Kustomisasi

3.9. Halaman Pesanan

Halaman pesanan, menampilkan daftar aktivitas transaksi seluruh *user* secara kronologis sebagaimana ditunjukkan pada Gambar 11. Pada gambar tersebut tercantum informasi tanggal pesanan, jam pesanan, produk yang dipesanan, nominal, *id* pesanan dan estimasi pembuatan/pengiriman.



Gambar 10. Halaman Pesanan

3.10. Blackbox Testing

Tabel 2. Blackbox testing

No	Fitur	Skenario Uji	Hasil	Status
1	Login	Input <i>email</i> dan <i>password</i> valid	Berhasil masuk ke dashboard	Valid
2	Login	Input <i>email</i> atau <i>password</i> salah	Muncul pesan error	Valid
3	Registrasi	Input data <i>user</i> lengkap dan valid	Akun berhasil dibuat	Valid
4	Registrasi	Input data tidak lengkap atau tidak valid	Muncul pesan validasi	Valid
5	Dashboard	Akses sebagai admin	Tampilan sesuai	Valid
6	Dashboard	Akses sebagai <i>user</i>	Informasi produk, kustomisasi, pesanan, chat	Valid
7	Pesanan	Bayar Sekarang	Berhasil dibayar	Valid
8	Pesanan	Batalan	Berhasil dibatalkan	Valid

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan sistem *e-commerce* elemen bangunan berbasis *web* pada UD. Wahyu Jati berhasil dirancang dan diimplementasikan sesuai dengan kebutuhan pengguna. Sistem yang dibangun mampu membantu proses digitalisasi penjualan yang sebelumnya masih dilakukan secara konvensional menjadi lebih efektif, terstruktur, dan terkomputerisasi. Melalui sistem ini, pelanggan dapat melihat katalog produk, melakukan kustomisasi produk, melakukan pemesanan, serta menyelesaikan pembayaran secara online tanpa harus datang langsung ke lokasi usaha.

Implementasi teknologi *Next.js* dan integrasi *payment gateway* *Midtrans* juga berhasil meningkatkan efisiensi proses transaksi dan kualitas pelayanan pelanggan. Sistem pembayaran digital yang terintegrasi memungkinkan transaksi dilakukan secara lebih cepat, aman, dan fleksibel melalui berbagai metode pembayaran seperti *transfer bank*, *virtual account*, *QRIS*, dan *e-wallet*. Selain itu, fitur kustomisasi produk memberikan kemudahan bagi pelanggan dalam menentukan spesifikasi produk sesuai kebutuhan sehingga mampu meningkatkan pengalaman pengguna dalam menggunakan sistem.

Berdasarkan hasil pengujian menggunakan metode *Black Box Testing*, seluruh fitur utama sistem dapat berjalan dengan baik sesuai dengan fungsi yang telah dirancang. Fitur *login*, *registrasi*, pengelolaan produk, kustomisasi, pemesanan, hingga pembayaran menunjukkan hasil valid tanpa ditemukan kesalahan fungsional yang signifikan. Dengan demikian, sistem *e-commerce* ini dinilai mampu membantu UD. Wahyu Jati dalam meningkatkan efektivitas operasional, memperluas jangkauan pemasaran, serta mendukung transformasi digital UMKM di era perkembangan teknologi informasi saat ini.

REFERENSI

- [1] M. Dody Firmansyah, "Perancangan Web E-Commerce Berbasis Website pada Toko Ida Shoes," 2023.
- [2] E. Nasri, E. Nuryani, and N. Agustiawan, "RANCANGAN E-COMMERCE DAN PAYMENT API MIDTRANS UNTUK PRODUK UMKM (STUDI KASUS PADA RUMAH KEMASAN)," *Jurnal Sistem Informasi dan Informatika (SIMIKA)*, vol. 4, 2021.
- [3] J. A. Stikom, U. Kupang, and N. T. Timur, "Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)," 2019. [Online]. Available: <http://ttskab.go.id/>
- [4] N. Pratiwi and A. Thariq, "E-COMMERCE INTEGRATED WITH PAYMENT GATEWAY USING EXTREME PROGRAMMING METHOD AT STARTUP YUKA KONVEKSI," Apr. 2025.
- [5] C. Gibran, A. R. Dewi, and E. Hadinata, "Implementasi Framework Laravel Untuk Pengembangan Website Penjualan Ayam Potong Dengan Pemanfaatan Midtrans Menggunakan Metode Fast," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 246–253, 2024.
- [6] W. A. Putra, I. Fitri, and D. Hidayatullah, "Implementasi Waterfall dan Agile dalam Perancangan E-commerce Alat Musik Berbasis Website," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 6, no. 1, p. 2022, 2022, doi: 10.35870/jti.