E-ISSN: 2829-0208

Jurnal Teknik Informatika dan Komputer

https://journal.uhamka.ac.id/index.php/jutikom

Pengembangan Sistem E-Commerce Toko Pakaian Berbasis Website Menggunakan ReactJS

Zakie Nurfaiz Ramadhan, Rahmi Imanda
Teknik Informatika, Fakultas Teknologi Industri dan Informatika Universitas Muhammadiyah Prof. DR. Hamka, Jakarta
Email: zakinurfaizramadhan@gmail.com, rahmi.imanda@uhamka.ac.id

Article Info	Received: August 19, 2024	Accepted: April 28, 2025	Published: September 31, 2025

ABSTRAK

Perkembangan teknologi mengubah pola belanja dari konvensional ke online, namun banyak pelaku usaha belum memiliki platform e-commerce yang efektif. Penelitian ini bertujuan mengembangkan sistem e-commerce toko pakaian berbasis website menggunakan ReactJS. Pengembangan sistem menggunakan metode Waterfall dengan tahapan perancangan, implementasi, dan pengujian. Perancangan menggunakan Use Case Diagram dan Sequence Diagram, sedangkan pengujian dilakukan dengan metode Black Box Testing. Hasil penelitian ini adalah platform e-commerce yang mempermudah transaksi, pengelolaan produk, dan manajemen pelanggan, sehingga meningkatkan efisiensi bisnis serta jangkauan pasar secara digital dengan baik dan efektif sekali, serta memberikan kemudahan bagi pelanggan dalam berbelanja online dengan lebih mudah dan cepat.

Kata kunci*: e-commerce, reactjs, waterfall, use case diagram, sequence diagram, blackbox testing*

ABSTRACT

Technological advancements have transformed shopping patterns from conventional to online, yet many businesses still lack an effective e-commerce platform. This research aims to develop a website-based e-commerce system for clothing stores using ReactJS. The system development employs the Waterfall method with stages of design, implementation, and testing. The design utilizes Use Case Diagrams and Sequence Diagrams, while testing is conducted using Black Box Testing methodology. The result of this research is an e-commerce platform that simplifies transactions, product management, and customer management, thereby increasing business efficiency and digital market reach effectively and efficiently. This platform also provides customers with a more convenient and faster online shopping experience.

Keywords: *e-commerce, reactjs, waterfall, use case diagram, sequence diagram, blackbox testing*

1. PENDAHULUAN

Perkembangan teknologi digital telah mendorong perubahan besar dalam dunia bisnis, termasuk sektor e-commerce yang semakin kompetitif (Sihombing, 2023). E-commerce memberikan kemudahan bagi konsumen dalam berbelanja secara online serta membuka peluang pasar yang lebih luas bagi pelaku usaha (Sartika, 2024). Di Indonesia, nilai transaksi e-commerce mencapai Rp 476,3 triliun pada tahun 2022 (Dataindonesia, 2023), yang menunjukkan peran besarnya dalam perekonomian digital. Faktor usia dan gender turut mempengaruhi pola konsumsi, di mana anak muda dan perempuan menjadi kelompok yang paling aktif dalam transaksi e-commerce (Abadi, 2020). Selain itu, lingkungan sosial, seperti pergaulan dan tren, juga berkontribusi dalam membentuk perilaku konsumsi individu, terutama di kalangan mahasiswa yang cenderung mengikuti gaya hidup yang sedang populer (Astuti, 2022).

Namun, efektivitas platform e-commerce sangat bergantung pada sistem yang mampu mengelola inventori, layanan pelanggan, dan proses transaksi secara efisien. Banyak toko pakaian berbasis e-commerce menghadapi kendala dalam mengelola stok ukuran dan warna secara real-time, serta kesulitan menyediakan layanan yang dipersonalisasi bagi pelanggan (Gunawan, 2019).

Idealnya, sistem e-commerce harus dapat mengoptimalkan pengelolaan stok, memberikan rekomendasi produk yang relevan, serta mengotomatisasi proses bisnis. Sayangnya, banyak platform masih menggunakan sistem terpisah untuk berbagai aspek tersebut, sehingga menyebabkan inefisiensi operasional dan pengalaman belanja yang kurang optimal (Utomo, 2022). Selain itu, internet yang berkembang pesat telah membawa perubahan besar dalam penyebaran informasi dan transaksi bisnis, yang semakin mendorong pertumbuhan pasar e-commerce (Sampurna, 2023).

Sebagai solusi, penelitian ini mengembangkan aplikasi e-commerce berbasis website untuk toko pakaian dengan menggunakan teknologi **ReactJS**. ReactJS merupakan pustaka **JavaScript open-source** yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna yang lebih cepat dan dinamis. Dengan pendekatan berbasis **microservices** dan **API RESTful**, sistem ini dapat meningkatkan skalabilitas, memungkinkan pengelolaan inventori secara real-time, serta mendukung integrasi dengan layanan pembayaran digital dan sistem rekomendasi produk. Studi terdahulu menunjukkan bahwa pendekatan ini dapat meningkatkan efisiensi operasional serta kualitas layanan dalam berbagai sektor digital (Rehatalanit, 2021). Selain itu, penggunaan **PHP: Hypertext Preprocessor** dalam pengembangan sistem memungkinkan pengelolaan data yang lebih efisien, terutama dalam pemrosesan transaksi digital (Purwanto, 2023).

Beberapa penelitian sebelumnya telah membahas pengembangan sistem berbasis web untuk **UKM** serta integrasi sistem pembayaran digital. Namun, penelitian sebelumnya lebih berfokus pada aspek individual, seperti sistem pembayaran atau manajemen stok, tanpa menghadirkan solusi terpadu (Ardiyansyah, 2022). Oleh karena itu, penelitian ini menawarkan pendekatan yang lebih menyeluruh dengan mengkombinasikan **pengelolaan inventori real-time**, **sistem rekomendasi produk berbasis data**, serta **integrasi pembayaran digital** dalam satu sistem yang lebih efisien dan dapat meningkatkan kepuasan pengguna (Gunawan, 2019).

Penelitian ini bertujuan untuk mengatasi tantangan dalam pengelolaan e-commerce untuk toko pakaian dengan membangun sistem yang lebih responsif, terintegrasi, dan efisien. Dengan pendekatan teknologi yang diterapkan, diharapkan sistem ini dapat meningkatkan efektivitas bisnis bagi pelaku usaha serta memberikan pengalaman belanja yang lebih baik bagi konsumen (Setiawan, 2018). Selain itu, penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan sistem e-commerce di masa depan dengan menyediakan **platform yang lebih cerdas, aman, dan adaptif terhadap perubahan pasar** (Alfarisi, 2024).

2. METODE PENELITIAN

Pada penelitian ini, metode Waterfall atau model **air terjun** digunakan dengan tahapan yang mencakup perancangan sistem, desain (use case diagram, sequence diagram, class diagram), implementasi website, serta pengujian sistem. Model ini menerapkan pendekatan sekuensial berbasis tahap, di mana setiap tahap harus diselesaikan sebelum berlanjut ke tahap berikutnya dalam pengembangan perangkat lunak (Maulida, 2022).

2.1. Teknik Perancangan

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuential atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (support) (Guring, 2024).

2.2. Teknik Pengujian

Metode Pengujian Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada fungsionalitas tanpa memeriksa kode program. Untuk memastikan apakah fungsi-fungsi dalam suatu perangkat lunak beroperasi dengan efektif, akurat, dan sesuai dengan spesifikasi yang telah ditetapkan, diperlukan pengujian terhadap website e-commerce ini (A. Ijudin, 2020). Selain itu, Black Box Testing juga dapat melakukan verifikasi serta validasi perangkat lunak yang sedang dibangun (S. Maji, 2021). Pengujian ini bertujuan untuk memastikan bahwa masukan dan keluaran sistem sesuai dengan persyaratan yang ditetapkan. Metode ini memiliki beberapa keunggulan, seperti tidak memerlukan keahlian teknis pemrograman bagi penguji, dapat mengungkap kelalaian dalam perangkat lunak yang mungkin ditemukan dari perspektif pengguna, membantu mengklarifikasi kontradiksi dalam eksekusi sistem, serta memungkinkan pengujian yang lebih cepat dibandingkan dengan metode white box.

3. HASIL DAN PEMBAHASAN

3.1. Perancangan Sistem

Perancangan sistem dalam pengembangan website e-commerce ini mengikuti tahapan metode Waterfall. Dimulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian. Untuk mendukung proses perancangan, digunakan Use Case Diagram dan Sequence Diagram sebagai alat bantu dalam memahami alur interaksi pengguna dengan sistem.

1. Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna dan sistem dalam berbagai skenario penggunaan.
2. Sequence Diagram digunakan untuk menggambarkan alur proses dalam sistem secara lebih rinci, termasuk urutan interaksi antar komponen.

Setelah perancangan selesai, sistem diimplementasikan dalam bentuk website e-commerce yang dapat mengelola transaksi penjualan, stok produk, serta data pelanggan secara lebih efisien. Tahapan akhir dalam pengembangan ini adalah pengujian sistem, yang dilakukan menggunakan metode Black Box Testing untuk memastikan bahwa

semua fungsi berjalan sesuai dengan spesifikasi yang telah ditetapkan.

3.2. Perancangan Use case Diagram

Use case ini menjelaskan interaksi yang terjadi dalam pembuatan Aplikasi Berbasis Website Penjualan Online Toko Pakaian yang tertera dalam tabel di bawah ini:

Tabel 1. Aktor Use case

No.	Aktor	Description
1.	*Admin*	Figur yang bisa mengelola atau manajemen data dalam sistem *website* penjualan online toko pakaian.
2.	*Customer*	Figur yang bisa melihat data produk, melakukan pemesanan dan dapat melakukan transaksi.
3.	*Visitor*	Figur yang hanya bisa melihat produk tetapi tidak bisa melakukan pemesanan.

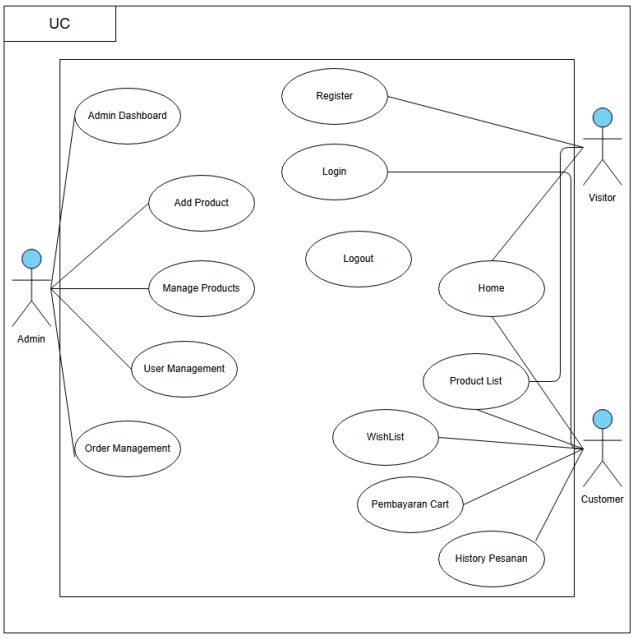
Use case Diagram

Diagram use case ini menggambarkan interaksi antara Admin dan Visitor dalam sistem yang diuraikan. Admin dapat mengakses fitur-fitur utama melalui Admin Dashboard, seperti mengelola produk, mengelola pengguna, dan mengelola pesanan. Fitur-fitur tersebut termasuk (include) dalam use case utama.

Di sisi lain, terdapat fitur-fitur lain yang dapat diakses oleh Visitor, seperti login, logout, melihat daftar produk, daftar keinginan, keranjang pembayaran, dan riwayat pesanan. Fitur-fitur ini memperluas (extend) fungsionalitas use case utama.

Visitor dapat melakukan registrasi untuk memperoleh akses ke sistem. Setelah login, Visitor dapat menjelajahi berbagai fitur yang tersedia, seperti melihat daftar produk, menambahkan ke daftar keinginan, dan melakukan pembayaran pesanan. Sementara Admin dapat mengelola semua aspek sistem melalui Admin Dashboard.

Secara keseluruhan, diagram use case ini mengilustrasikan pembagian peran dan fungsionalitas antara Admin dan Visitor dalam sistem yang dijelaskan.

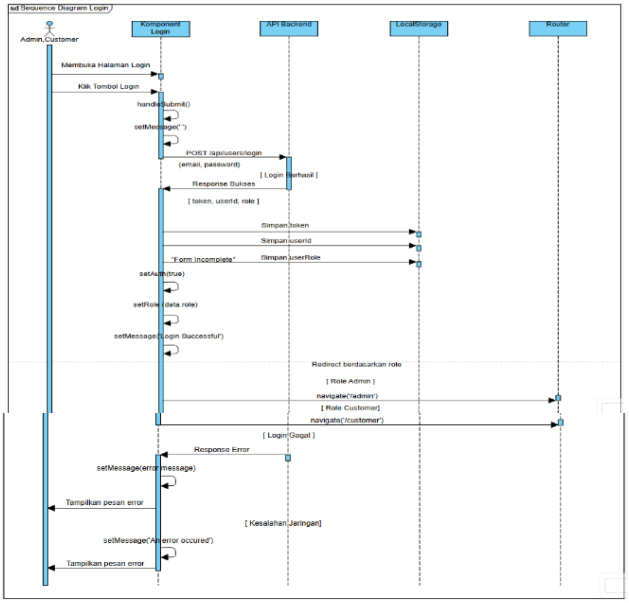


Gambar 1. Use case diagram

3.3. Perancangan Sequence Diagram

Sequence Diagram Login

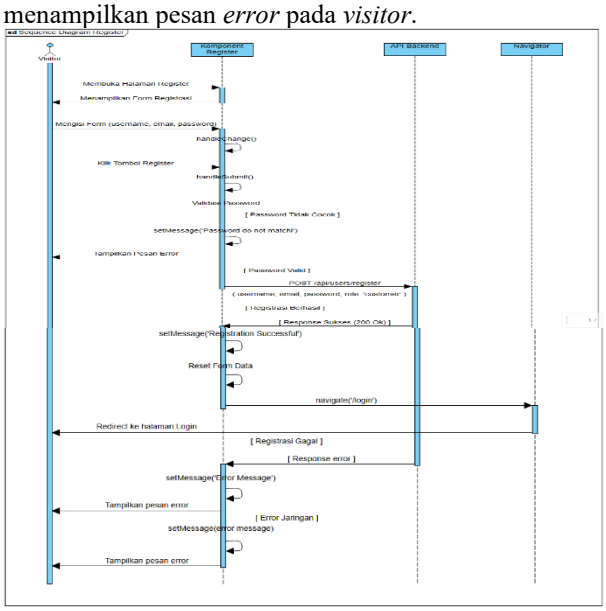
Alur *login* dimulai ketika *Admin/Customer* membuka halaman *login*. Setelah itu *user* mengklik tombol *login* yang akan memicu *handleSubmit()* untuk mengirimkan data (*email*, *password*) ke *API Backend* melalui POST /api/users/login. Jika *login* berhasil, *API* akan mengembalikan *response* berupa *token*, *userId* dan *role* yang kemudian disimpan ke *LocalStorage*. Sistem akan melakukan *redirect* berdasarkan *role* - jika *admin* akan diarahkan ke '/admin' dan jika *customer* ke '/customer'. Jika login gagal, sistem akan menampilkan pesan *error* pada *user*.



Gambar 2. Sequence diagram untuk proses login

Sequence Diagram Register

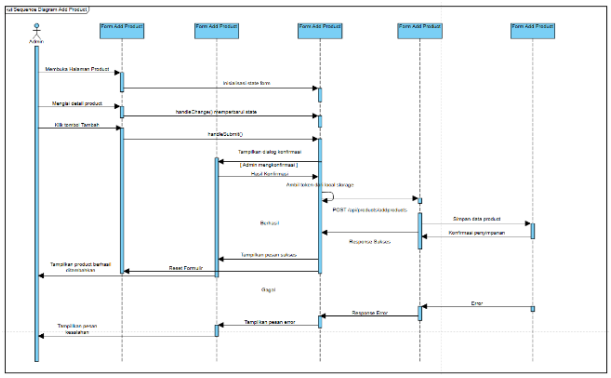
Alur registrasi dimulai ketika *Visitor* membuka halaman *register* dan menampilkan form registrasi. *Visitor* mengisi form dengan memasukkan *username*, *email* dan *password* yang akan memicu *handleChange()* untuk validasi input. Setelah mengklik tombol *register*, sistem akan memicu *handleSubmit()* dan melakukan validasi *password*. Jika *password* valid, data registrasi (*username, email, password, role: 'customer'*) akan dikirim ke API Backend melalui POST /api/users/register. Jika registrasi berhasil, sistem akan menampilkan pesan sukses dan melakukan *redirect* ke halaman *login*. Namun jika terjadi kegagalan baik karena *password* tidak cocok atau *error* jaringan, sistem akan menampilkan pesan *error* pada *visitor*.



Gambar 3. Sequence diagram untuk proses register

Sequence Diagram Tambah Product

Alur penambahan produk dimulai ketika *Admin* membuka halaman *product*. Sistem akan menginisialisasi state form untuk pengisian detail produk. Setelah *Admin* mengisi detail produk, sistem akan memperbarui *state* melalui *handleChange()*. Ketika *Admin* mengklik tombol Tambah, sistem memicu *handleSubmit()* dan menampilkan dialog konfirmasi. Setelah *Admin* mengkonfirmasi, sistem akan mengambil *token* dari *local storage* dan mengirim data produk ke API Backend melalui POST /api/products/addproducts. Jika berhasil, sistem akan menyimpan data produk, menampilkan pesan sukses, dan mereset formulir. Namun jika gagal, sistem akan menampilkan pesan kesalahan kepada *Admin*.



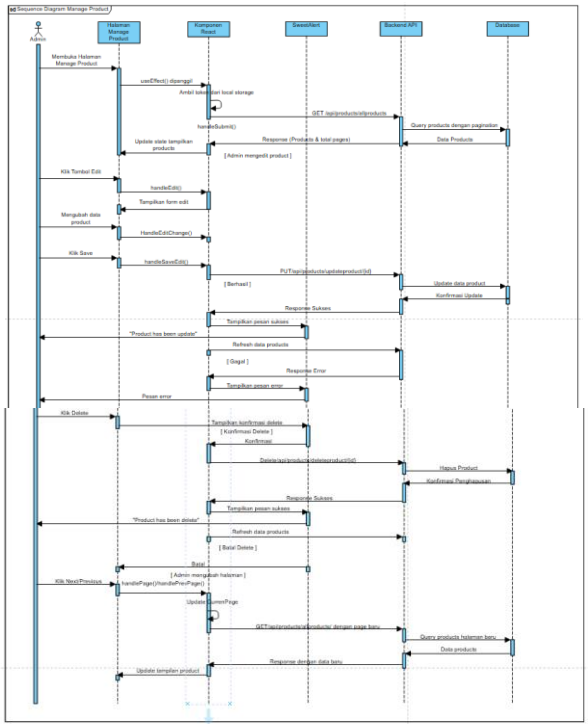
Gambar 4. Sequence diagram proses tambah product

Sequence Diagram Mengelola Product

Alur *manage product* dimulai saat *Admin* membuka halaman *Manage Product*. Sistem akan mengambil *token* dari *local storage* dan melakukan GET /api/products/allproducts untuk *query* produk dengan *pagination* dari *database*. Data produk ditampilkan dan *Admin* dapat melakukan beberapa aksi:

1. *Edit*: *Admin* dapat mengklik tombol *Edit* untuk menampilkan form *edit*, mengubah data produk melalui *handleEditChange()*, dan menyimpan perubahan dengan *handleSaveEdit()* yang akan mengirim PUT request ke API. Jika berhasil data akan diperbarui.
2. *Delete*: Admin dapat mengklik tombol *Delete* yang akan menampilkan konfirmasi. Setelah dikonfirmasi, sistem akan mengirim DELETE request ke API untuk menghapus produk.
3. *Navigasi*: *Admin* dapat mengklik *Next/Previous* untuk melihat halaman produk lainnya melalui *handlePage()/handlePrevPage()* yang akan memperbarui tampilan dengan data produk halaman baru.

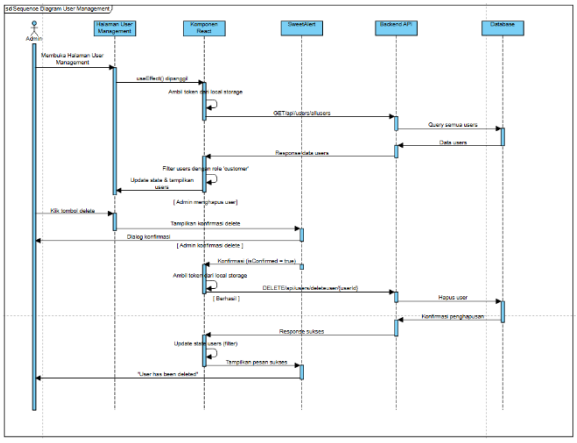
Setiap aksi yang gagal akan menampilkan pesan *error* pada *Admin.*



Gambar 5. Sequence diagram proses mengelola product

Sequence Diagram Mengelola Pengguna

Alur *user management* dimulai ketika *Admin* membuka halaman *User Management*. Sistem mengambil *token* dari *local storage* dan melakukan GET /api/users/allusers untuk *query* semua *users*. Data *users* di filter berdasarkan *role 'customer'* dan ditampilkan. *Admin* dapat menghapus *user* dengan mengklik tombol *delete* yang akan menampilkan konfirmasi. Setelah dikonfirmasi, sistem mengambil *token* dan mengirim DELETE *request*. Jika berhasil, daftar *users* di*update* dan pesan sukses ditampilkan.

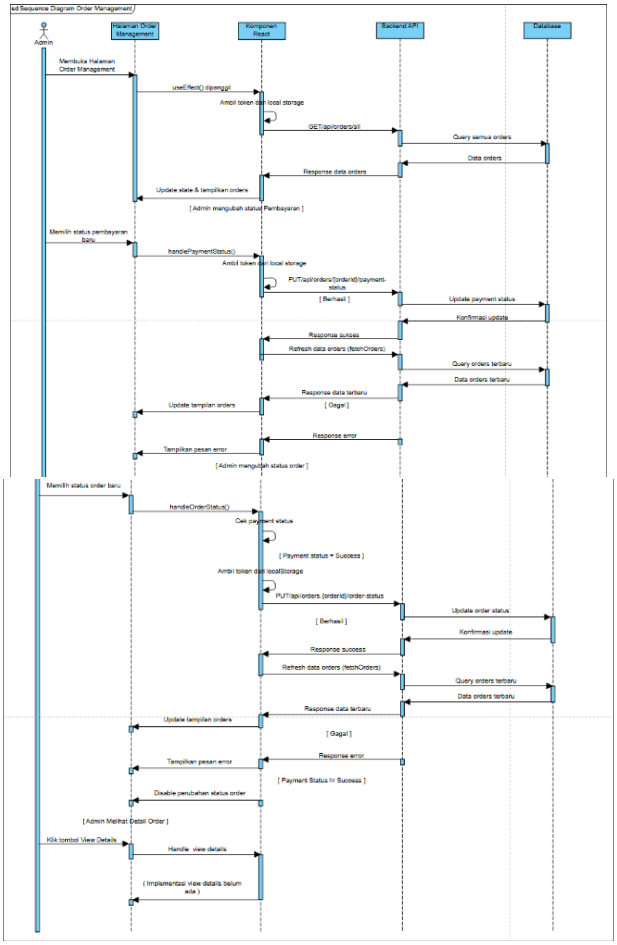


Gambar 6. Sequence diagram mengelola pengguna

Sequence Diagram Mengelola Pesanan

Alur *order management* dimulai saat *Admin* membuka halaman *Order Management*. Sistem mengambil *token* dari *local storage* dan melakukan GET /api/orders/all untuk *query* semua *orders* dari *database*. Data *orders* ditampilkan dan *Admin* dapat:

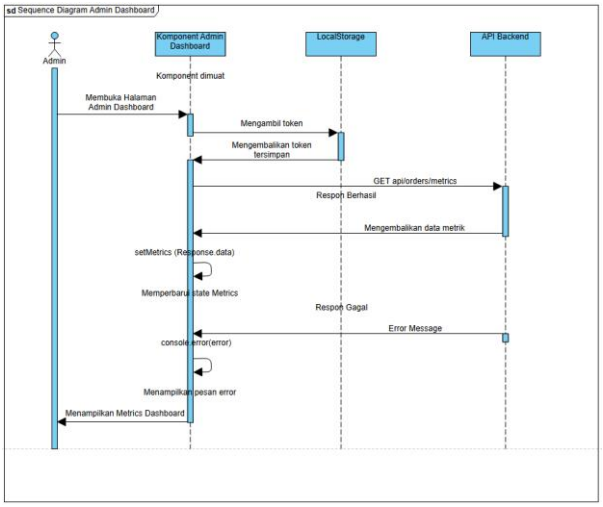
1. Mengubah status pembayaran: *Admin* memilih status pembayaran baru yang memicu *handlePaymentStatus()*, mengambil *token* dan mengirim PUT *request* untuk *update* status pembayaran. Jika berhasil, data *orders* di*refresh*.
2. Mengubah status *order*: *Admin* memilih status *order* baru yang memicu *handleOrderStatus()*, sistem cek *payment* status. Jika *success*, *token* diambil dan PUT request dikirim untuk *update* status *order*. Setelah berhasil, data *orders* di*refresh*.
3. Melihat *detail*: *Admin* dapat klik *View Details* untuk melihat *detail order* (implementasi belum ada).



Gambar 7. Sequence diagram mengelola pesanan

Sequence Diagram Mengelola Dashboard Admin

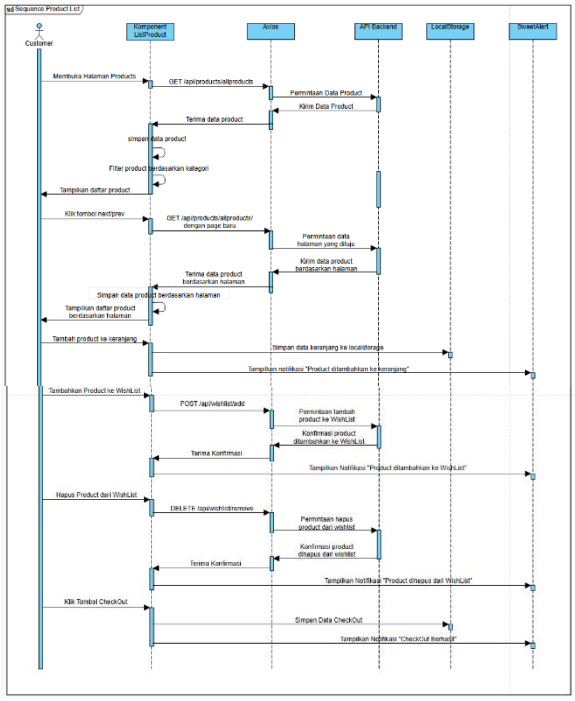
Alur *admin dashboard* dimulai ketika *Admin* membuka halaman *Admin Dashboard*. Sistem akan mengambil *token* dari *LocalStorage* dan mengembalikan *token* tersimpan. Kemudian sistem melakukan GET *request* ke api/orders/metrics untuk mendapatkan data metrik. Jika berhasil, sistem akan mengatur metrik menggunakan setMetrics(Response.data) dan memperbarui state Metrics untuk menampilkan *Metrics Dashboard*. Jika gagal, sistem akan menampilkan pesan *error* melalui *console.error(error)*.



Gambar 8. Sequence diagram mengelola dashboard admin

Sequence Diagram Daftar Product

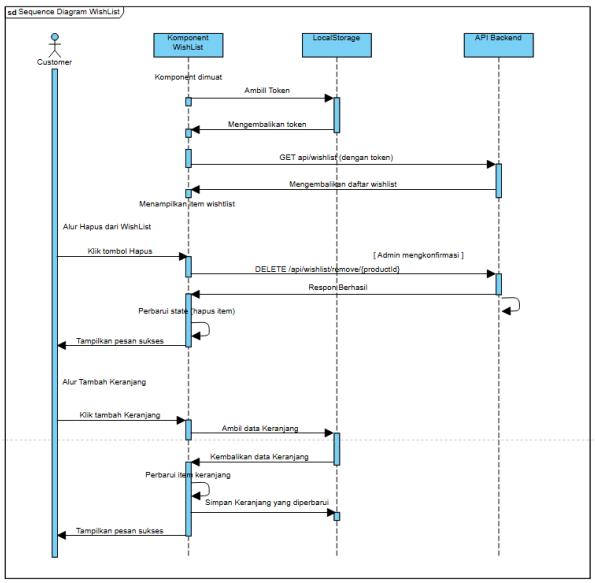
Alur *Product List* dimulai ketika *Customer* membuka halaman *Products* yang memicu GET *request* ke /api/products/allproducts melalui komponen *ListProduct*. Data produk yang diterima dari API Backend kemudian disimpan dan dapat di filter berdasarkan kategori sebelum ditampilkan ke *Customer*. *Customer* dapat melakukan navigasi antar halaman menggunakan tombol *next/prev* yang akan memicu GET *request* baru dengan parameter halaman yang dituju. *Customer* memiliki opsi untuk menambahkan produk ke keranjang yang akan menyimpan data ke *localStorage* dan menampilkan notifikasi "*Product* ditambahkan ke keranjang", atau menambahkan ke *WishList*.



Gambar 9. Sequence diagram daftar product

Sequence Diagram WishList

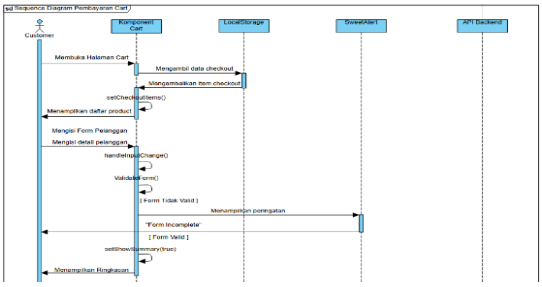
Alur *WishList* dimulai saat komponen *WishList* dimuat, sistem pertama mengambil token dari *localStorage* untuk autentikasi. Setelah mendapatkan *token*, sistem melakukan GET request ke /api/wishlist untuk mengambil daftar *wishlist* yang kemudian ditampilkan ke *Customer*. *Customer* dapat melakukan dua aksi utama: menghapus item dari *WishList* yang akan mengirim DELETE *request* ke /api/wishlist/remove/{productId} atau menambahkan item ke keranjang. Setelah item berhasil dihapus atau ditambahkan ke keranjang, sistem akan memperbarui tampilan dan menampilkan pesan sukses kepada *Customer*.

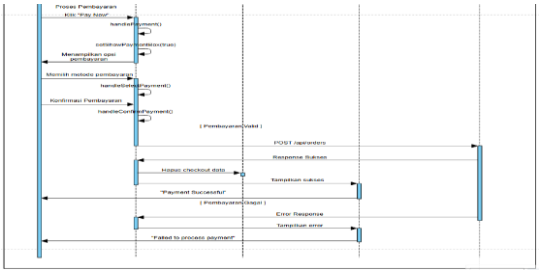


Gambar 10. Sequence diagram mengelola wishlist

Sequence Diagram Pembayaran Cart

Alur pembayaran *cart* dimulai saat *Customer* membuka halaman *cart* dimana sistem akan mengambil data *checkout* dari *localStorage* dan menampilkannya. *Customer* kemudian memilih daftar produk dan mengisi form detail pelanggan yang akan divalidasi oleh sistem. Jika form tidak *valid*, sistem menampilkan peringatan *"Form Incomplete"*, namun jika *valid* sistem menampilkan ringkasan pembayaran. *Customer* dapat melakukan pembayaran dengan mengklik *"Pay Now"*, memilih metode pembayaran, dan melakukan konfirmasi. Sistem kemudian mengirim POST *request* ke /api/orders - jika sukses akan menampilkan *"Payment Successful"* dan menghapus data *checkout*, jika gagal akan menampilkan pesan *"Failed to process payment"*.

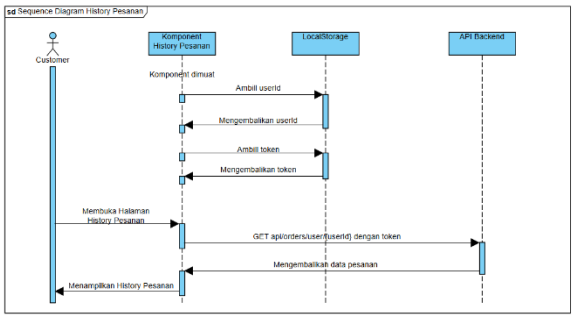




Gambar 11. Sequence diagram pembayaran cart

Sequence Diagram Riwayat Pesanan

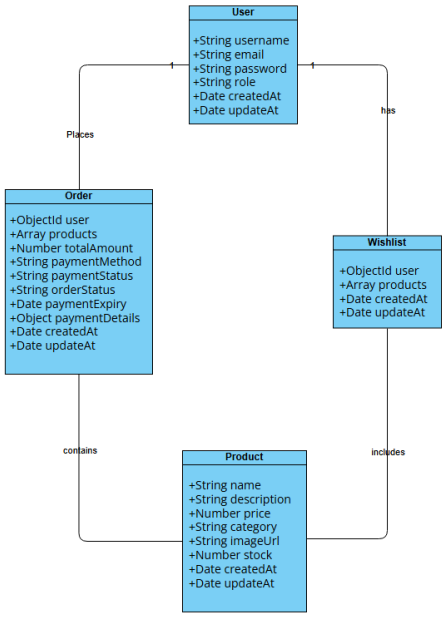
Alur *History* Pesanan dimulai ketika komponen *History* Pesanan dimuat, dimana sistem pertama-tama mengambil *userId* dari *localStorage* dan kemudian mengambil token untuk autentikasi. Setelah *Customer* membuka halaman *History* Pesanan, sistem melakukan GET request ke /api/orders/user/{userId} dengan menyertakan *token* untuk otentikasi. API Backend kemudian mengembalikan data pesanan berdasarkan *userId* tersebut, dan sistem menampilkan *history* pesanan kepada *Customer* melalui komponen *History* Pesanan.



Gambar 12. Sequence diagram melihat riwayat pesanan

3.4. Perancangan Class Diagram

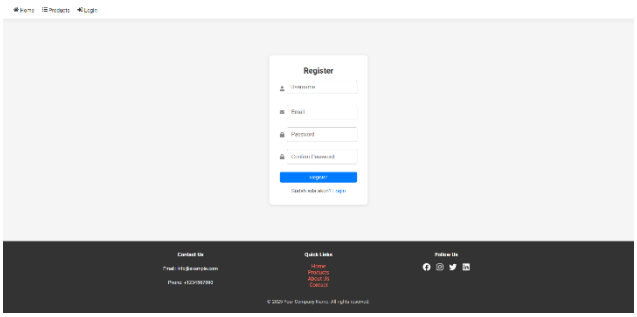
Dalam alur e-commerce ini, seorang User dapat mendaftar dan membuat akun dengan peran customer atau admin. Setelah terdaftar, User dapat menjelajahi berbagai Product yang tersedia, menambahkan Product ke Wishlist sebagai daftar keinginan, dan kemudian membuat Order dengan memilih satu atau beberapa Produk yang ingin dibeli. Setiap Order memiliki detail pembayaran dan status, mulai dari pending hingga delivered, dengan metode pembayaran yang dapat dipilih dan status pembayaran yang dapat berubah dari pending menjadi success atau cancelled, mencerminkan perjalanan transaksi dari awal hingga selesai dalam sistem e-commerce ini.



Gambar 13. Class Diagram

Hasil Implementasi Register

Halaman *Register* memungkinkan pengguna untuk membuat akun baru dengan mudah. Pengguna diminta mengisi beberapa informasi dasar, seperti nama pengguna, email, dan kata sandi. Setelah semua data diisi dengan benar, mereka dapat menekan tombol *Register* untuk menyelesaikan proses registrasi. Jika sudah memiliki akun, pengguna dapat beralih ke halaman *Login* melalui tautan yang tersedia.



Gambar 14. Tampilan Halaman Form Register

3.5. Implementasi Sistem

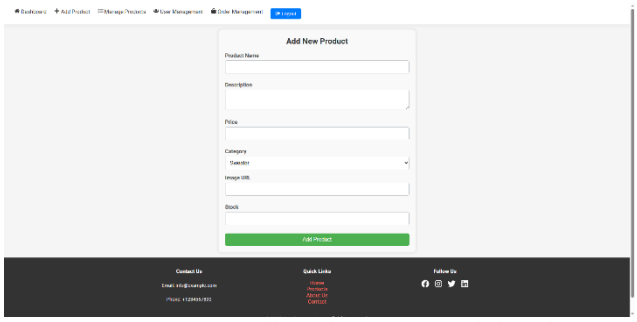
Hasil Implementasi Login

Halaman *Login* dirancang untuk memudahkan pengguna dalam mengakses akun mereka. Pengguna hanya perlu memasukkan email dan kata sandi yang telah terdaftar, lalu menekan tombol *Login* untuk melanjutkan. Jika pengguna belum memiliki akun, terdapat tautan untuk melakukan pendaftaran.

Gambar 15. Tampilan Halaman Form Login

Hasil Implementasi Tambah Product

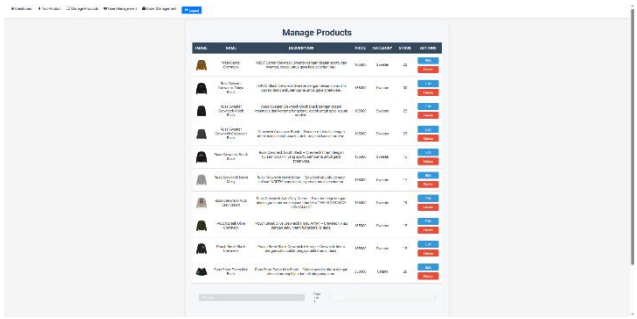
Halaman *Add Product* dirancang untuk memungkinkan *admin* menambahkan produk baru ke dalam sistem dengan mudah. Admin dapat mengisi informasi produk seperti nama, harga, kategori, deskripsi, serta mengunggah gambar produk. Setelah semua data diisi, admin dapat menekan tombol *"Add Product"* untuk menyimpan produk ke dalam database. Proses ini memastikan bahwa setiap produk yang ditambahkan memiliki informasi yang lengkap dan akurat sebelum ditampilkan kepada pelanggan.



Gambar 16. Tampilan Halaman Form Tambah Product

Hasil Implementasi Mengelola Product

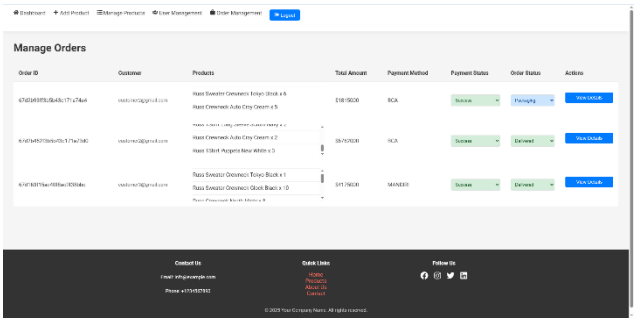
Halaman *Manage Products* berfungsi sebagai pusat pengelolaan produk dalam sistem. *Admin* dapat melihat daftar produk yang telah ditambahkan, serta mengedit atau menghapus produk yang sudah ada. Dengan fitur ini, *admin* dapat dengan mudah memperbarui informasi produk atau menghapus produk yang tidak lagi tersedia, memastikan data produk selalu terkini dan relevan bagi pelanggan.



Gambar 17. Tampilan Halaman Mengelola Product

Hasil Implementasi Mengelola Pesanan

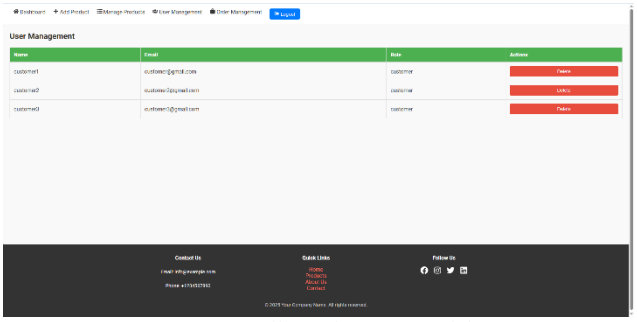
Halaman *Order Management* memungkinkan admin untuk mengelola pesanan pelanggan secara efisien. Setiap pesanan yang masuk ditampilkan dengan detail seperti nama pelanggan, produk yang dipesan, total harga, dan status pesanan. Admin dapat memperbarui status pesanan dari "Menunggu Pembayaran" hingga "Dikirim" untuk memastikan pelanggan mendapatkan informasi terkini mengenai pesanan mereka. Dengan fitur ini, proses pemantauan dan pengelolaan pesanan menjadi lebih sistematis dan terorganisir.



Gambar 18. Tampilan Halaman Mengelola Pesanan Pelanggan

Hasil Implementasi Mengelola Pengguna

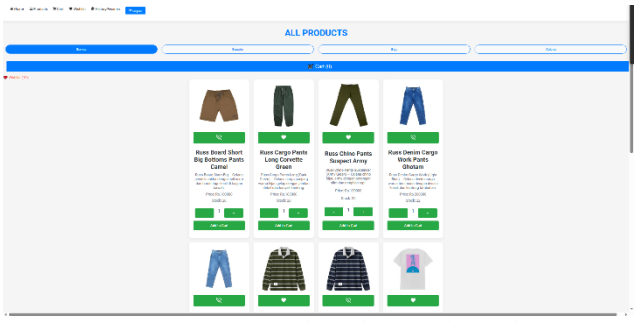
Halaman *User Management* digunakan untuk mengelola data pengguna dalam sistem. Admin dapat melihat daftar pelanggan yang terdaftar, menghapus akun pengguna yang tidak aktif. Fitur ini membantu dalam menjaga keamanan dan keteraturan sistem dengan memastikan hanya pengguna yang valid yang dapat mengakses fitur tertentu dalam aplikasi.



Gambar 19. Tampilan Halaman Mengelola Pengguna

Hasil Implementasi Daftar Product

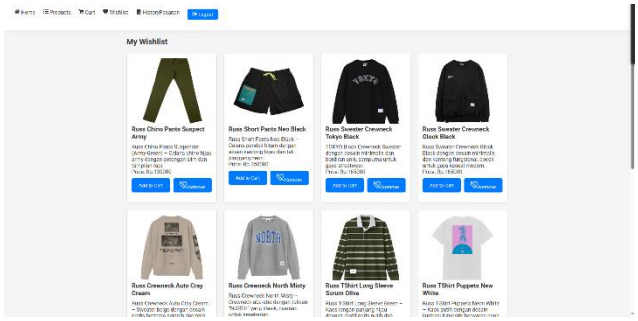
Halaman *List Products* menampilkan daftar lengkap produk yang tersedia dalam sistem. Setiap produk disertai dengan informasi seperti nama, harga, deskripsi, dan gambar untuk memudahkan pelanggan dalam memilih barang yang sesuai dengan kebutuhan mereka. Fitur pencarian dan filter juga dapat digunakan untuk mempercepat proses pencarian produk tertentu.



Gambar 20. Tampilan Halaman Daftar Product

Hasil Implementasi WishList

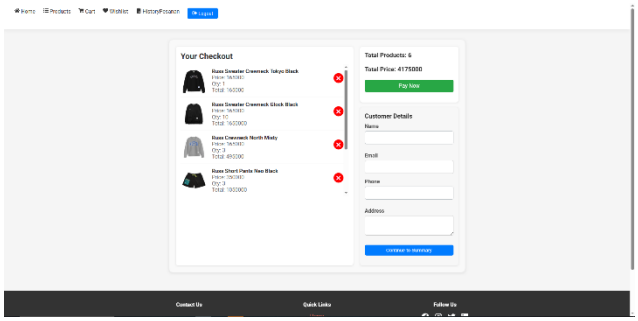
Halaman *WishList* memungkinkan pengguna untuk menyimpan produk favorit mereka untuk dibeli di lain waktu. Dengan fitur ini, pelanggan dapat dengan mudah menandai produk yang menarik minat mereka tanpa perlu mencarinya kembali. *WishList* membantu meningkatkan pengalaman belanja dengan memberikan fleksibilitas dalam perencanaan pembelian.



Gambar 21. Tampilan Halaman WishList

Hasil Implementasi Pembayaran Cart

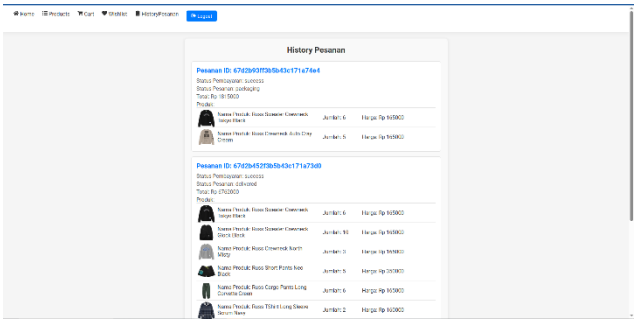
Halaman *Cart* memungkinkan pelanggan untuk meninjau produk yang telah mereka pilih sebelum melakukan pembelian. Pengguna dapat menyesuaikan jumlah produk, menghapus item yang tidak diinginkan, dan melihat total harga secara real-time. Fitur ini membantu memastikan bahwa pelanggan memiliki kontrol penuh atas pesanan mereka sebelum melanjutkan ke proses pembayaran.



Gambar 22. Tampilan Halaman Pembayaran Cart

Hasil Implementasi Riwayat Pesanan

Halaman *History* Pesanan menyediakan riwayat transaksi pelanggan, menampilkan status pesanan dari tahap pembayaran hingga pengiriman. Dengan fitur ini, pelanggan dapat dengan mudah melacak pesanan mereka, melihat detail pembelian sebelumnya, dan memastikan pesanan diterima sesuai dengan harapan.



Gambar 23. Tampilan Halaman Riwayat Pesanan

3.6. Pengujian Sistem

Black Box Testing

Untuk mengetahui apakah perangkat lunak berfungsi sesuai yang diharapkan, dilakukan pengujian untuk memastikan bahwa fungsionalitas perangkat lunak terjamin.

Tabel 2. Hasil Pengujian Autentikasi

No	Funsi diuji	Scenario Pengujian	Hasil yang diharapkan	Hasil
1.	Proses Register	1.Pengguna membuka halaman Register. 2. Mengisi data (email, password, dsb.) dengan benar. 3. Menekan tombol Register.	Akun baru berhasil dibuat dan dapat digunakan untuk Login.	OK
2.	Proses Register Email sama	1. Pengguna membuka halaman Register. 2. Mengisi data dengan email yang sudah terdaftar. 3. Menekan tombol Register.	Sistem menampilkan pesan kesalahan (“Email sudah terdaftar”) dan mencegah pembuatan akun baru.	OK
3.	Proses Login	1. Pengguna memasukkan email dan password yang valid. 2. Menekan tombol Login.	Berhasil masuk ke dalam sistem (dashboard/home).	OK
4.	Proses Login dengan Email/Password salah	1. Pengguna memasukkan email/password yang tidak sesuai. 2. Menekan tombol Login.	Sistem menampilkan pesan error (“Email atau password salah”) dan tidak diizinkan untuk login.	OK

Tabel 3. Hasil Pengujian Fitur

No	Fungsi yang diuji	Scenario Pengujian	Hasil yang diharapkan	Hasil
1.	Tambah Product (Admin)	1.Admin membuka halaman Add Product. 2. Mengisi form produk (nama, deskripsi, harga, kategori, gambar, stok). 3.Menekan tombol "Add Product". 4.Sistem menampilkan pesan konfirmasi bahwa produk berhasil ditambahkan.	Data produk baru berhasil ditambahkan ke dalam sistem dan muncul di halaman Manage Products.	OK
2.	Mengelola Product (Admin)	1.Admin membuka halaman Manage Product. 2.Untuk menghapus produk: Admin menekan tombol "Delete" pada produk yang dipilih, lalu mengonfirmasi penghapusan. 3.Untuk mengedit produk: Admin menekan tombol "Edit", mengubah data produk, dan menekan tombol simpan. 4.Sistem menampilkan pesan konfirmasi atas penghapusan atau pembaruan data produk.	Produk berhasil dihapus dan data produk yang diedit diperbarui sesuai dengan input yang diberikan.	OK
3.	Mengelola Pengguna (Admin)	1.Admin membuka halaman User Management. 2.Admin memilih akun pengguna yang tidak aktif. 3.Menekan tombol "Delete" untuk menghapus akun tersebut. 4.Sistem meminta konfirmasi sebelum menghapus. 5.Setelah konfirmasi, sistem menghapus akun pengguna dan menampilkan notifikasi penghapusan.	Akun pengguna yang tidak aktif berhasil dihapus, sehingga sistem hanya menampilkan pengguna yang valid.	OK
4.	Mengelola Pesanan (Admin)	1.Admin membuka halaman Order Management. 2.Memilih pesanan yang ingin diperbarui statusnya. 3.Mengubah status pembayaran dan status pesanan melalui dropdown atau tombol yang tersedia. 4.Menekan tombol simpan/submit. 5.Menekan opsi "View Details" untuk melihat rincian pesanan. 6.Sistem menampilkan status terbaru dan detail pesanan sesuai dengan perubahan yang dilakukan.	Status pembayaran dan status pesanan berhasil diperbarui, dan detail pesanan dapat dilihat dengan jelas.	OK
5.	Daftar Product (Custo-mer)	1.Pelanggan membuka halaman Product List. 2.Memilih kategori produk (misalnya baju, celana, sweater). 3.Menekan ikon love product	Pelanggan dapat melihat produk sesuai kategori, menambahkan produk ke wishlist, dan melakukan checkout	OK

		untuk menambahkan produk ke wishlist. 4.Memilih produk yang diinginkan dan menekan tombol "add cart" yang tersedia di halaman tersebut. 6.Sistem memasukkan produk yang dipilih ke dalam pembayaran cart.	langsung dari halaman product list sehingga produk masuk ke dalam pembayaran cart.	
6.	Pemba-yaran Cart (Custo-mer)	1.Pelanggan membuka halaman Pembayaran Cart. 2.Pelanggan menghapus produk yang tidak ingin dibeli dari keranjang. 3. Mengisi form pembayaran dengan data yang diperlukan (nama, email, alamat). 4.Menekan tombol "Bayar" dan melakukan transfer sesuai instruksi. 5.Sistem memproses pembayaran dan menampilkan konfirmasi bahwa pembayaran berhasil.	Produk yang dihapus tidak muncul dalam proses pembayaran dan pembayaran berhasil diproses dengan konfirmasi yang jelas.	OK
7.	Riwayat Pesanan (Custo-mer)	1.Pelanggan membuka halaman History Pesanan. 2.Melihat daftar pesanan yang telah dibuat. 3.Memeriksa status setiap pesanan (sudah dikirim atau masih dalam proses). 4.Jika diinginkan, pelanggan dapat mengklik pesanan untuk melihat detail lebih lanjut. 5.Sistem menampilkan informasi lengkap dan status pengiriman dari setiap pesanan.	Riwayat pesanan tampil dengan informasi status yang akurat, memudahkan pelanggan memantau transaksi mereka.	OK

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa:

1. Pengembangan sistem e-commerce berbasis website untuk toko pakaian telah berhasil dirancang menggunakan metode Waterfall. Tahapan pengembangannya mencakup perancangan sistem dengan alat bantu Use Case Diagram dan Sequence Diagram, implementasi website menggunakan ReactJS, serta pengujian sistem dengan metode Black Box Testing untuk memastikan fungsionalitas sistem berjalan sesuai spesifikasi yang ditetapkan.
2. Sistem yang dikembangkan memiliki tiga aktor utama dengan peran yang terdefinisi dengan jelas:
   a. Admin: Mengelola produk, pengguna, dan pesanan melalui dashboard admin.
   b. Customer: Dapat melihat produk, melakukan pemesanan, dan menyelesaikan transaksi.
   c. Visitor: Hanya dapat melihat produk tanpa dapat melakukan pemesanan.
3. Implementasi fitur-fitur utama sistem meliputi:
   a. Manajemen produk (CRUD operations).
   b. Sistem autentikasi (login/register).
   c. Wishlist dan keranjang belanja.
   d. Sistem pembayaran yang terintegrasi.
   e. Manajemen pesanan dengan status tracking.
   f. Riwayat pesanan bagi pelanggan.

Penelitian ini membuktikan bahwa pengembangan website e-commerce dengan pendekatan sistematis dapat meningkatkan efisiensi dalam pengelolaan produk, transaksi, dan layanan pelanggan, serta memberikan pengalaman belanja yang lebih baik bagi pengguna.

DAFTAR PUSTAKA

A. Ijudin, A. S. (2020). Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis. Jurnal Informatika Universitas Pamulang, 1, 8-12.

Abadi, A. F. (2020). Studi Perilaku Konsumtif Pada Mahasiswa Pendidikan Akuntansi. Jurnal Benefita, 5(2), 264-274.

Alfarisi, R. R. (2024). Pengembangan Aplikasi Web E-Commerce dan Donasi (Studi Kasus: Green

Welfare Indonesia). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 8(7).

Ardiyansyah, R. &. (2022). Sistem Informasi Penjualan Daging Menerapkan Model User Centered Design Berbasis Web. Jurnal Media Informatika Budidarma, 6(2), 760. doi:https://doi.org/10.30865/mib.v6i2.3562

Astuti, R. F. (2022). Pengaruh Modernitas dan Gaya Hidup terhadap Perilaku Konsumsi Mahasiswa. Jurnal Pendidikan Ekonomi Undiksha, 14(2), 237-245.

Dataindonesia. (2023, January 23). Transaksi e-Commerce RI Tak Capai Target pada 2022. Retrieved from Data Indonesia: https://dataindonesia.id/digital/detail/transaksi-ecommerce-ri-tak-capai-target-pada-2022

Gunawan, F. a. (2019). Analysis of the effects of perceived ease of use and perceived usefulness on consumer attitude and their impacts on purchase decision on PT Tokopedia in Jabodetabek. European Journal of Business and Management Research, 4.

Guring, D. S. (2024, 02 (February)). Perancangan Sistem Informasi Data Pelanggan Berbasis Web Pada Indogrosir Karawang. Jurnal Komputer dan Teknologi, 3(2), 43-48. Retrieved from Ngulikode: https://www.ngulikode.com/2023/02/panduan-pemula-untuk-menggunakan-fetch.html

Maulida, N. H. (2022). Studi literatur penerapan metode prototype dan waterfall dalam pembuatan sebuah aplikasi atau website. Universitas Palangkaraya. Palangkaraya: Universitas Palangkaraya.

Nursaid, F. F. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus: Toko Uda Fajri). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 46--55.

Pratama, F. R. (2020). Pengembangan Aplikasi E-Commerce Menggunakan Payment Gateway Midtrans. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 4, 1133--1140.

Purwanto, D. D. (2023). Pengembangan Aplikasi Human Resource Management pada PT. HJMB Menggunakan JS, React Native, dan GraphQL. Journal of Information System, Graphics, Hospitality and Technology, 5(2), 95-101. doi:https://doi.org/10.37823/insight.v5i2.337

Rehatalanit, Y. (2021). Peran e-commerce dalam pengembangan bisnis. Jurnal Teknologi Industri, 5.

Rosa, A. S. (2016). Rekayasa Perangkat Lunak Terstruktur Berorientasi Objek. Bandung: Informatika.

S. Maji, J. F. (2021, October). What is in the black box? – A perspective on software in cryoelectron microscopy. Biophysical Journal, 120(20), 4307-4311. doi:10.1016/j.bpj.2021.09.015

Sampurna, A. P. (2023). Pembangunan Aplikasi Website E-Commerce Dalam Peningkatan Penjualan Produk Perusahaan PT. Bala Biotech Indonesia. Journal Of Informatic Pelita Nusantara, 2, 433-438.

Sartika, D. U. (2024, January 23). Fenomena Penggunaan E-Commerce terhadap Perilaku Konsumsi Mahasiswa. WISSEN: Jurnal Ilmu Sosial dan Humaniora, 2(3), 335-350. Retrieved from ANTARA News: https://www.antaranews.com/berita/3360933/rudiantara-e-commerce-terus-kuat-dan-jadi-penopang-ekonomi-ri-2023

Setiawan, D. (2018). Dampak Perkembangan Teknologi Informasi dan Komunikasi Terhadap Budaya. Jurnal Simbolika : Research and Learning in Communication Study, 4(1), 62-72.

Sihombing, R. A. (2023). Evaluasi Usability Aplikasi Shopee pada Proses Pembelian Online Dengan Metode User Centered Design. Jurnal SIFO Mikroskil, 24(2), 81-94. doi:https://doi.org/10.55601/jsm.v24i2.1023

Utomo, B. T. (2022). Membangun Website E-Commerce Spare Part Yamaha Max's Garage Semarang Dengan Menggunakan Metode UCD (User Centered Design). Google Scholar, 7(Sens 7).