

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Penelitian ini menggunakan pendekatan kualitatif dengan tujuan memahami kebutuhan, permasalahan, serta konteks operasional UMKM Labuhanbatu secara mendalam sebelum merancang solusi teknologi. Teknik pengumpulan data yang digunakan adalah:

3.1.1 Wawancara

Wawancara dilakukan secara daring dengan pelaku UMKM di Labuhanbatu untuk mengetahui:

1. Proses bisnis penjualan produk yang sedang berjalan.
2. Kendala dalam pemasaran produk secara *online*.
3. Fitur-fitur yang diinginkan pada *website* penjualan.

Wawancara ini dilakukan melalui *Google Form* dengan daftar pertanyaan yang telah disusun sebelumnya untuk memudahkan responden memberikan jawaban yang terstruktur.

3.1.2 Survei (Kuesioner)

Survei dilakukan untuk mengetahui preferensi dan kebutuhan pengguna terhadap *website* penjualan *online*. Kuesioner disebarakan kepada calon pengguna melalui tautan *Google Form*.

1. Tingkat minat konsumen terhadap belanja *online*.
2. Kebutuhan fitur pada *website* (misalnya, katalog produk, keranjang belanja).

3. Persepsi kemudahan penggunaan *website* (*user experience*)

3.1.3 Studi Pustaka

Buku, jurnal, dan dokumentasi resmi terkait pengembangan *web* dengan *CodeIgniter*, rekayasa perangkat lunak, *UML /DFD*, perancangan basis data relasional, dan standar kualitas perangkat lunak.

3.2 Metode Perancangan Sistem

Perancangan sistem dilakukan menggunakan model *Design & Development Research* yang dipadukan dengan penekatan rekayasa perangkat lunak berulang (*iteratif*) dan *MVC* (*Model-View-Controller*) pada *CodeIgniter*. Tahapan utamanya:

1. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan langkah awal dalam proses perancangan sistem yang bertujuan untuk menggali, memahami, dan mendefinisikan kebutuhan pengguna serta permasalahan yang hendak diselesaikan. Pada tahap ini, dilakukan pengumpulan informasi melalui observasi, wawancara, kuesioner, atau studi literatur terkait proses bisnis pengguna, dalam hal ini pelaku UMKM. Data yang diperoleh dianalisis untuk mengidentifikasi kebutuhan fungsional (fungsi utama sistem) dan *non-fungsional* (keamanan, kinerja, dan keandalan).

2. Perancangan Sistem (arsitektur, proses, data, antarmuka)

Perancangan sistem dilakukan dengan membangun rancangan arsitektur sistem, alur proses, model data, serta rancangan antarmuka pengguna. Rancangan arsitektur menggambarkan komponen utama sistem, pola komunikasi antar-modul, serta teknologi yang digunakan. Model proses dapat digambarkan melalui

diagram alir, *UML (Use Case Diagram, Activity Diagram)*, atau *Data Flow Diagram (DFD)* untuk menunjukkan aliran data dan proses utama. Perancangan data dilakukan dengan membangun *Entity Relationship Diagram (ERD)* atau model basis data yang sesuai dengan kebutuhan. Sementara itu, desain antarmuka berfokus pada pembuatan tampilan (*mockup* atau *wireframe*) yang mudah digunakan (*user-friendly*) dan sesuai dengan prinsip *UX/UI* untuk meningkatkan kenyamanan pengguna.

3. Implementasi menggunakan *CodeIgniter*

Tahap implementasi adalah proses mengubah desain sistem menjadi kode program yang dapat dijalankan. *Framework CodeIgniter* digunakan sebagai *platform* pengembangan karena sifatnya yang ringan, cepat, serta mendukung pola arsitektur *MVC (Model-View-Controller)* yang memudahkan pemisahan logika, tampilan, dan data. Implementasi dimulai dengan pengaturan struktur folder, konfigurasi *database*, pembuatan model untuk mengakses data, pembuatan *controller* untuk mengatur logika bisnis, dan pembuatan *view* sebagai antarmuka pengguna.

4. Uji Coba Internal (*developer*)

Setelah implementasi selesai, dilakukan tahap uji coba bagian dalam (*internal testing*) oleh tim pengembang untuk memastikan semua modul berjalan sesuai rancangan dan bebas dari kesalahan teknis (*bug*).

5. *Refinement*

Tahap *refinement* dilakukan sebagai tindak lanjut dari hasil uji coba. Setiap kesalahan dianalisis untuk melakukan perbaikan sistem, baik pada aspek antarmuka, fungsionalitas, maupun performa. Proses ini bersifat iteratif, artinya

perbaikan dapat dilakukan secara berulang hingga sistem benar-benar sesuai dengan kebutuhan pengguna. Selain itu, pada tahap ini juga dapat ditambahkan fitur tambahan atau peningkatan keamanan jika diperlukan, sehingga sistem menjadi lebih optimal dan siap untuk diterapkan secara penuh oleh UMKM.

3.2.1 Rancangan Masukan (*Input*)

Rancangan masukan berfungsi sebagai titik awal interaksi pengguna dengan sistem. Masukan adalah data atau informasi yang dimasukkan oleh pengguna ke dalam sistem untuk diproses lebih lanjut. Pada sistem yang dirancang menggunakan *CodeIgniter* dan ditujukan bagi pelaku UMKM, rancangan *input* dirancang agar sederhana, efisien, dan mudah dipahami oleh pengguna *non*-teknis.

Desain *input* dilakukan dengan mempertimbangkan tiga aspek utama: jenis data yang dimasukkan, metode *input*, dan validasi *input*. Jenis data yang dimasukkan bisa berupa teks (nama produk, deskripsi), angka (harga, jumlah stok), tanggal (tanggal transaksi), dan *file* (*upload* gambar produk). Metode *input* dirancang melalui formulir isian (form *input*) yang terstruktur dan intuitif, menggunakan elemen *HTML*.

Setiap *input* difasilitasi dengan validasi data baik di sisi klien (*client-side*) menggunakan *JavaScript* maupun di sisi server (*server-side*) melalui fitur *built-in CodeIgniter*. Validasi ini mencakup pengecekan format, batas panjang karakter, keunikan data (seperti *ID* pengguna), dan keharusan pengisian (*required fields*). Rancangan masukan juga memperhatikan aspek *usability*, seperti pemberian label yang jelas, petunjuk pengisian, dan *feedback* saat terjadi kesalahan pengisian.

3.2.1.1 Rancangan Masukan Data Pengguna

Rancangan ini berfungsi sebagai *form input* untuk mengelola akun pengguna di *website* UMKM, di mana data yang dimasukkan oleh pengguna, seperti *email*, nama, dan *password*, akan disimpan di basis data untuk keperluan autentikasi dan identifikasi pengguna. Tampilan ini dibuat sederhana dan jelas, agar memudahkan pengguna, baik admin, pelanggan maupun pemilik UMKM, dalam melakukan penambahan pengguna (*user*), pendaftaran akun baru dan *login*. Desain ini sangat penting sebagai bagian dari sistem *user authentication* yang Menjadi pintu masuk utama untuk mengakses layanan atau fitur di *website* UMKM.

The image displays three wireframe forms for user management on the UMKM website:

- WEBSITE UMKM (Login Form):** Features a header with the website name, input fields for 'Email' and 'Passwo', a 'Login' button, and a link 'Belum Punya akun?' at the bottom.
- Register Pelanggan Baru (Register Form):** Features a title, input fields for 'Nama Lengkap', 'Email', and 'Password', a 'Daftar' button, and a link 'Sudah punya akun? Login' at the bottom.
- Tambah User (Add User Form):** Features a title, input fields for 'Nama', 'Email', 'Password', and 'Role', a 'Simpan' button, and a link 'Kembali' at the bottom.

Gambar 3.1 *Form register, Form Login, Form Tambah User*

3.2.1.2 Rancangan Masukan Data Produk

Setiap data yang dimasukkan akan diproses dan disimpan dalam *database* sebagai informasi produk yang ditampilkan kepada pelanggan. Rancangan ini dirancang sederhana, fokus pada kemudahan pengisian data, serta menampilkan elemen-elemen penting yang dibutuhkan untuk deskripsi produk, harga, dan

ketersediaan stok. Formulir ini merupakan bagian krusial dari fitur manajemen produk pada *website* UMKM, memungkinkan pelaku usaha untuk menambah produk baru secara efisien dan cepat.



The image shows a web form titled "Tambah" (Add) for adding a new product. It contains the following fields and a button:

- Nama Produk**: A single-line text input field.
- Deskripsi Produk**: A multi-line text area for product description.
- Harga (Rp)**: A single-line text input field for the price.
- Stok**: A single-line text input field for the stock quantity.
- Foto Produk**: A single-line text input field, likely for a file upload path or name.
- Tambah Produk**: A rounded rectangular button at the bottom to submit the form.

Gambar 3. 2 Tampilan *Input* Produk

Tampilan ini terdiri dari beberapa kolom *input* yang digunakan oleh pemilik UMKM untuk menambahkan informasi produk yang akan dijual. Kolom *input* yang tersedia meliputi Nama Produk, Deskripsi Produk, Harga (Rp), Stok, serta fitur unggah foto produk melalui tombol *Choose File*. Di bagian bawah, terdapat tombol Tambah Produk yang berfungsi untuk mengirim data produk ke sistem.

3.2.1.3 Rancangan Masukan Data Transaksi

Rancangan masukan (*input*) untuk sistem *e-commerce* pada *website* UMKM, di mana keranjang belanja berfungsi untuk menampung daftar produk yang dipilih oleh pelanggan sebelum melakukan pembelian. *Input* pada halaman ini berupa jumlah produk yang dapat diubah, serta aksi hapus yang mempengaruhi data dalam keranjang. Informasi subtotal dan total harga menjadi *output* dari data yang sudah di-*input*-kan. Halaman ini dirancang agar pelanggan dapat dengan

mudah mengelola produk yang ingin dibeli, sekaligus menjadi tahap awal dalam proses *checkout* dan transaksi pada *website* UMKM.

Keranjang Belanja Saya					
No	Nama Produk	Harga	Jumlah	Subtotal	Aksi
1		Rp. -		Rp. -	Hapus
2		Rp. -		Rp. -	Hapus
				Total: Rp.	
<div style="text-align: center; margin-bottom: 5px;"> <input type="button" value="Checkout Sekarang"/> </div> <div style="text-align: center;"> Kembali ke Katalog Produk </div>					

Gambar 3. 3 Tampilan *input* keranjang belanja

Di bagian atas terdapat tabel yang menampilkan daftar produk yang telah dipilih oleh pelanggan, meliputi kolom No, Nama Produk, Harga, *UML*, Subtotal, dan Aksi. Setiap baris produk memiliki tombol hapus (*Delete*) pada kolom aksi, yang memungkinkan pengguna untuk menghapus produk dari keranjang. Di bagian bawah tabel, terdapat tombol bertuliskan *Checkout* Sekarang untuk melanjutkan ke proses pembayaran. Selain itu, tersedia tautan “Kembali ke Katalog Produk” yang berfungsi untuk kembali melihat produk lainnya.

3.2.2 Rancangan Proses (*Process*)

Rancangan proses merupakan tahapan perencanaan alur kerja sistem yang menjelaskan bagaimana data yang masuk akan diolah menjadi keluaran yang bermanfaat sesuai tujuan sistem. Proses ini dirancang dengan mengacu pada hasil analisis kebutuhan dan spesifikasi sistem yang telah disusun sebelumnya.

Secara umum, rancangan proses digambarkan melalui diagram alir data untuk memperjelas hubungan antara aktor, data, dan alur aktivitas. Proses utama meliputi pengelolaan produk (tambah, ubah, hapus, dan lihat), pengelolaan transaksi penjualan, pengolahan data pelanggan serta manajemen pesanan. Setiap proses didesain mengikuti pola arsitektur *MVC* yang diimplementasikan di *CodeIgniter*, di mana *Model* menangani interaksi dengan basis data, *Controller* mengatur logika bisnis, dan *View* menampilkan hasil kepada pengguna.

Dalam perancangannya, setiap proses juga dilengkapi dengan mekanisme kontrol seperti autentikasi dan otorisasi pengguna, pengecekan validitas data, serta penanganan kesalahan (*error handling*).

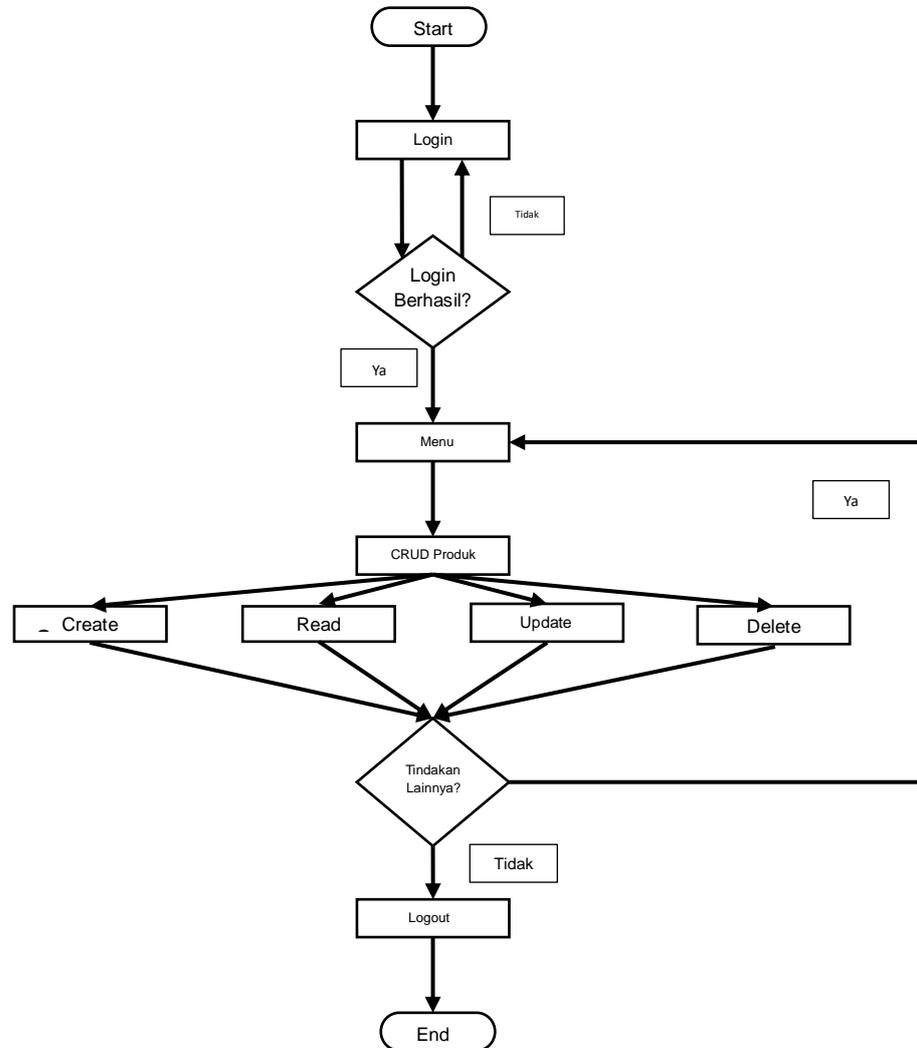
1. Manajemen Produk (*CRUD* produk)

Proses dimulai ketika pengguna mengakses sistem seperti pada diagram alur di bawah ini, ditandai dengan simbol oval bertuliskan "*Start*". Langkah pertama yang dilakukan oleh pengguna adalah *login* ke dalam sistem. Proses *login* ini ditampilkan dalam bentuk persegi panjang, yang merepresentasikan aktivitas atau tindakan yang dilakukan oleh pengguna.

Setelah memasukkan data untuk *login*, sistem akan melakukan validasi terhadap data yang dimasukkan. Tahapan ini menunjukkan bahwa sistem perlu mengambil keputusan, yakni apakah *login* berhasil atau tidak. Jika *login* berhasil, pengguna akan diarahkan ke langkah selanjutnya. Namun, jika *login* gagal, pengguna akan dikembalikan ke proses *login* untuk mencoba kembali.

Setelah berhasil *login*, pengguna akan diarahkan ke Menu Utama, di mana mereka dapat memilih berbagai opsi pengelolaan sistem. Salah satu opsi utama dalam menu ini adalah mengelola produk atau stok. Pengguna kemudian memilih

opsi tindakan spesifik, di mana salah satu fitur utama adalah *CRUD* Produk (*Create, Read, Update, Delete*).



Gambar 3. 4 Flowchart manajemen produk

Dalam tahap *CRUD* Produk, pengguna dapat melakukan:

- a. *Create*: Menambahkan produk baru ke dalam sistem.
- b. *Read*: Melihat daftar produk yang telah tersedia.
- c. *Update*: Mengedit atau memperbaiki informasi produk.

Setelah pengguna menyelesaikan salah satu tindakan tersebut, sistem akan menanyakan kembali melalui keputusan: Apakah pengguna ingin melakukan

tindakan lain?. Jika jawabannya "Ya", maka pengguna akan diarahkan kembali ke Menu Utama untuk melakukan aktivitas lainnya. Namun jika jawabannya "Tidak", maka sistem akan melanjutkan ke proses berikutnya.

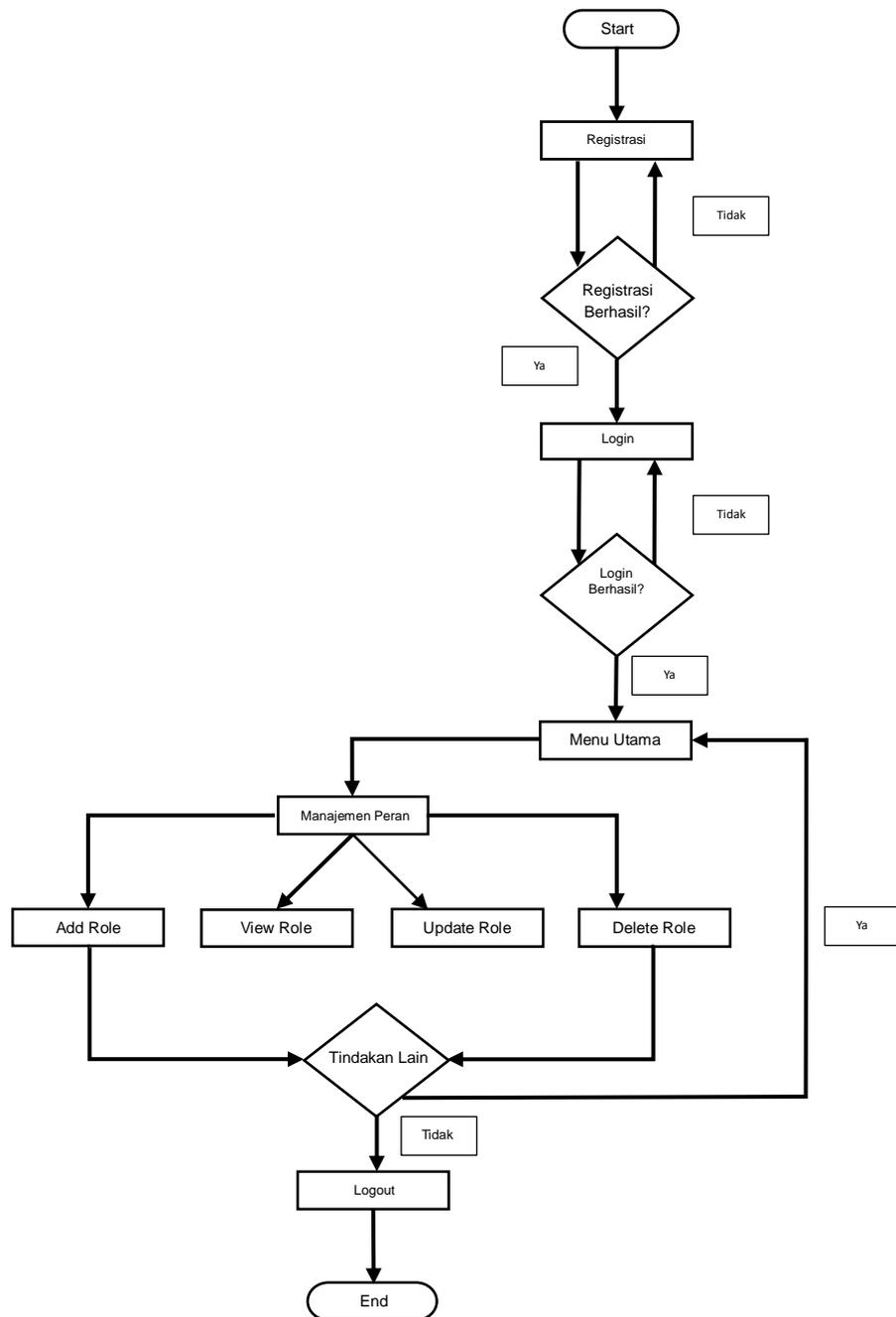
Langkah terakhir dari proses ini adalah *Logout*, di mana pengguna keluar dari sistem, dan kemudian alur sistem ditutup dengan simbol oval yang bertuliskan "*End*", yang menandakan bahwa seluruh proses telah selesai.

Alur ini direpresentasikan dengan beberapa simbol utama, yaitu:

- a. Oval digunakan untuk menandai titik awal (*Start*) dan akhir (*End*) dari proses.
 - b. Persegi panjang mewakili berbagai tindakan atau proses yang dilakukan pengguna
 - c. Belah ketupat menunjukkan titik pengambilan keputusan.
 - d. Panah mengarahkan alur logis dari satu langkah ke langkah berikutnya.
2. Manajemen Pengguna (registrasi, *login*, manajemen peran)

Alur sistem dimulai dengan simbol oval bertuliskan "*Start*", yang menunjukkan titik awal dari proses interaksi pengguna dengan sistem. Langkah pertama yang dilakukan adalah proses registrasi, khusus bagi pengguna baru yang belum memiliki akun.

Setelah pengguna mengisi formulir registrasi, sistem akan memverifikasi data yang dimasukkan. Pada tahap ini, terdapat keputusan yang penting, yaitu: Apakah registrasi berhasil? Jika registrasi berhasil, pengguna diarahkan ke proses *login*. Jika tidak, pengguna dikembalikan ke proses registrasi untuk mengulangi langkahnya.



Gambar 3.5 Flowchart Manajemen Pengguna

Pada proses *login*, pengguna yang sudah terdaftar memasukkan informasi *login* mereka untuk mengakses sistem. Setelah *login* dilakukan, sistem akan kembali melakukan verifikasi dan mengambil keputusan: Apakah login berhasil?.

Jika berhasil, maka pengguna akan diarahkan ke menu utama. Jika gagal, pengguna diarahkan kembali ke proses *login*.

Di dalam Menu Utama, pengguna diberikan berbagai pilihan untuk mengakses fitur sistem, salah satunya adalah fitur Manajemen Peran.

Dalam Manajemen Peran, pengguna dapat melakukan beberapa aktivitas utama terkait pengelolaan peran (*role*), yaitu:

- a. *Add Role*: Menambahkan peran baru ke dalam sistem.
- b. *View Roles*: Melihat daftar peran yang telah ada.
- c. *Update Role*: Mengubah informasi dari peran yang sudah tersedia.
- d. *Delete Role*: Menghapus peran tertentu dari sistem.

Setelah pengguna menyelesaikan salah satu atau beberapa tindakan tersebut, sistem akan memberikan pilihan melalui keputusan: Apakah pengguna ingin melakukan tindakan lain? Jika jawabannya "Ya", maka alur akan kembali ke Menu Utama sehingga pengguna bisa memilih opsi lain. Jika jawabannya "Tidak", maka sistem akan melanjutkan ke tahap *Logout*, di mana pengguna keluar dari sistem.

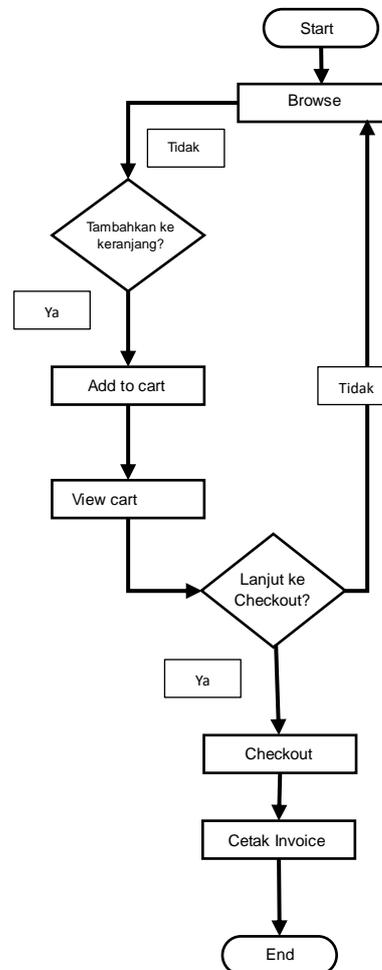
Langkah terakhir ditandai dengan simbol oval bertuliskan "*End*", yang menandakan bahwa seluruh proses interaksi sistem telah selesai.

Adapun simbol-simbol alur yang digunakan dalam sistem ini adalah sebagai berikut:

- a. Oval: Menandai titik awal (*Start*) dan akhir (*End*) dari proses sistem.
- b. Persegi Panjang: Mewakili proses atau aktivitas pengguna dalam sistem.
- c. Belah Ketupat: Menyimbolkan titik pengambilan keputusan.

- d. Panah: Menghubungkan setiap tahapan dan menunjukkan arah alur logika dari satu proses ke proses lainnya.

3. Proses Pemesanan



Gambar 3. 6 Flowchart proses pemesanan

Proses ini dimulai ketika pengguna mengakses sistem atau *platform e-commerce*, yang ditandai dengan simbol oval bertuliskan "Start". Setelah masuk ke halaman utama, pengguna dapat mulai melakukan aktivitas *Browse*, yaitu menjelajahi berbagai produk yang tersedia di sistem.

Setelah menjelajahi produk, sistem memberikan pilihan kepada pengguna melalui keputusan: Apakah pengguna ingin menambahkan produk ke keranjang?

Jika pengguna memilih "Ya", maka proses dilanjutkan ke langkah berikutnya, yaitu *Add to Cart*. Namun, jika pengguna menjawab "Tidak", maka alur kembali ke langkah *Browse*, memungkinkan pengguna untuk terus menjelajahi produk lainnya.

Pada proses *Add to Cart*, pengguna menambahkan satu atau beberapa produk ke dalam keranjang belanja. Setelah produk berhasil dimasukkan ke keranjang, pengguna dapat melanjutkan ke langkah berikutnya, yaitu *View Cart*, untuk melihat isi keranjang yang telah mereka kumpulkan.

Setelah melihat isi keranjang, pengguna akan dihadapkan pada keputusan selanjutnya: Apakah ingin melanjutkan ke *Checkout*? Jika pengguna memilih "Ya", maka sistem akan mengarahkan ke proses *Checkout*, di mana pengguna melengkapi informasi yang dibutuhkan untuk menyelesaikan pembelian. Namun, jika pengguna belum ingin menyelesaikan transaksi, maka alur kembali diarahkan ke proses *Browse*, sehingga pengguna bisa terus menambahkan produk lainnya ke dalam keranjang.

Langkah terakhir adalah *Checkout*, yang merupakan proses penting dalam menyelesaikan transaksi pembelian. Setelah proses ini selesai, alur sistem berakhir dengan simbol oval bertuliskan "End", yang menunjukkan bahwa keseluruhan siklus interaksi pengguna dalam berbelanja telah selesai.

Penjelasan Simbol Alur Sistem:

- a. Oval: Digunakan untuk menunjukkan titik awal (*Start*) dan akhir (*End*) dari alur sistem.
- b. Persegi Panjang: Menandai proses atau aktivitas yang dilakukan oleh pengguna

- c. Belah Ketupat: Menunjukkan titik pengambilan keputusan
- d. Panah: Menggambarkan arah alur proses, menunjukkan transisi logis antar langkah yang terjadi dalam sistem.

3.2.2.1 Diagram Konteks

Diagram Konteks memodelkan sistem sebagai satu proses tunggal dengan entitas eksternal:

1. Admin

Bertanggung jawab untuk meng-*input* dan mengelola data pengguna (*users*) dalam sistem. Admin akan mengirimkan data pengguna ke sistem dan menerima konfirmasi bahwa data telah berhasil disimpan

Aliran data:

- a. *Input*: Data *user* baru (misalnya akun Pemilik UMKM atau Pelanggan).
- b. *Output*: Konfirmasi pendaftaran atau *update* data pengguna.

2. Pemilik UMKM

Pemilik UMKM merupakan pengguna utama sistem yang memiliki fitur-fitur sebagai berikut:

- a. Mendaftar dan *login* ke dalam sistem.
- b. Mengelola produk (*input* data produk, edit, hapus).
- c. Mengelola pesanan yang dibuat oleh pelanggan.

Aliran data:

- a. *Input*: Data pendaftaran, *login*, data produk, *update* pesanan, permintaan laporan.
- b. *Output*: Daftar produk, data pesanan dari pelanggan, laporan transaksi.

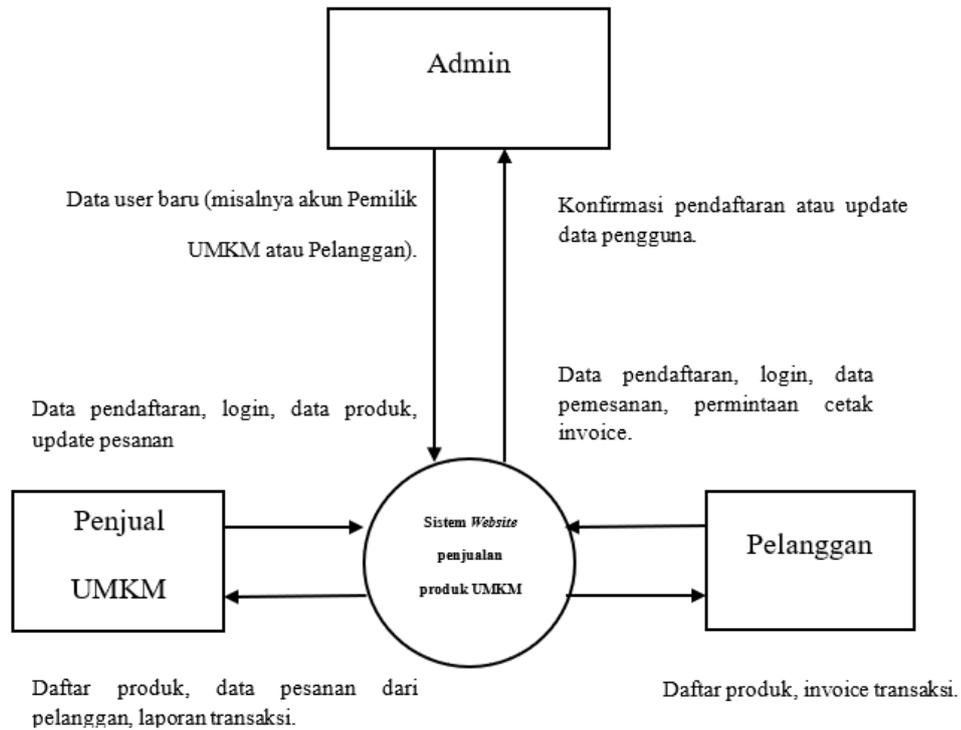
3. Pelanggan

Pelanggan adalah pengguna akhir yang menggunakan sistem untuk membeli produk.

- a. Mendaftar dan *login* ke dalam sistem.
- b. Memilih produk dari daftar yang ditampilkan.
- c. Melakukan pemesanan produk yang diinginkan.
- d. Mencetak *invoice* dan mengirim *invoice* ke Pemilik UMKM.

Aliran data:

- a. *Input*: Data pendaftaran, *login* , data pemesanan, permintaan cetak *invoice*.
- b. *Output*: Daftar produk, *invoice* transaksi.



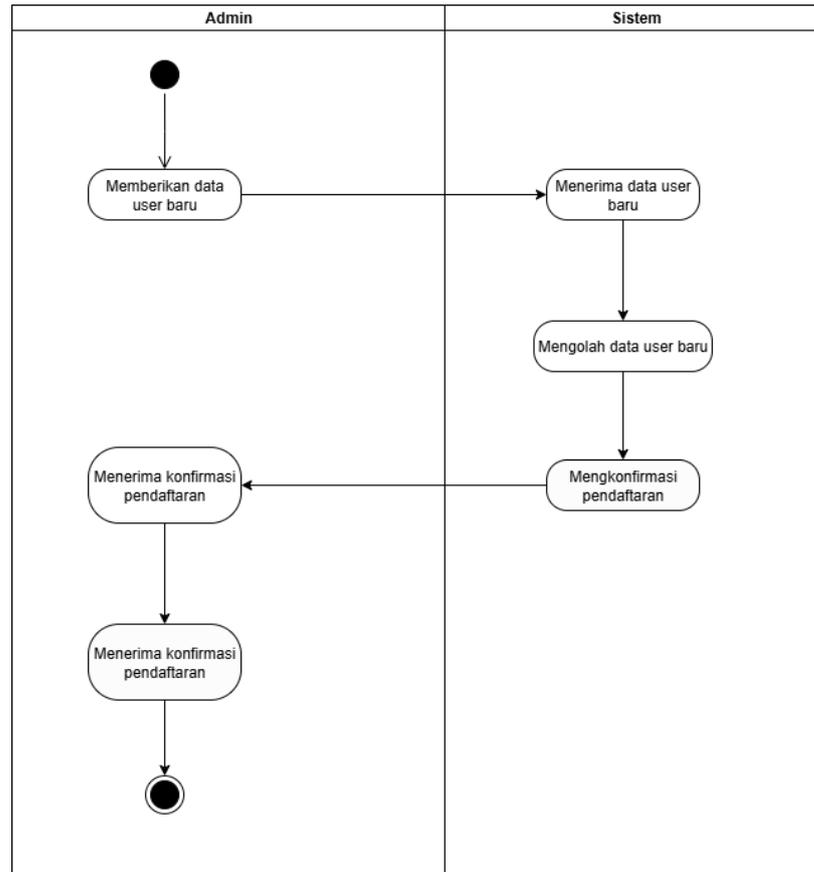
Gambar 3. 7 Diagram Konteks *Web* UMKM

3.2.2.2 *Activity Diagram*

Unified Modeling Language (UML) digunakan untuk memodelkan alur aktivitas.

1. *Activity Diagram* Proses Registrasi

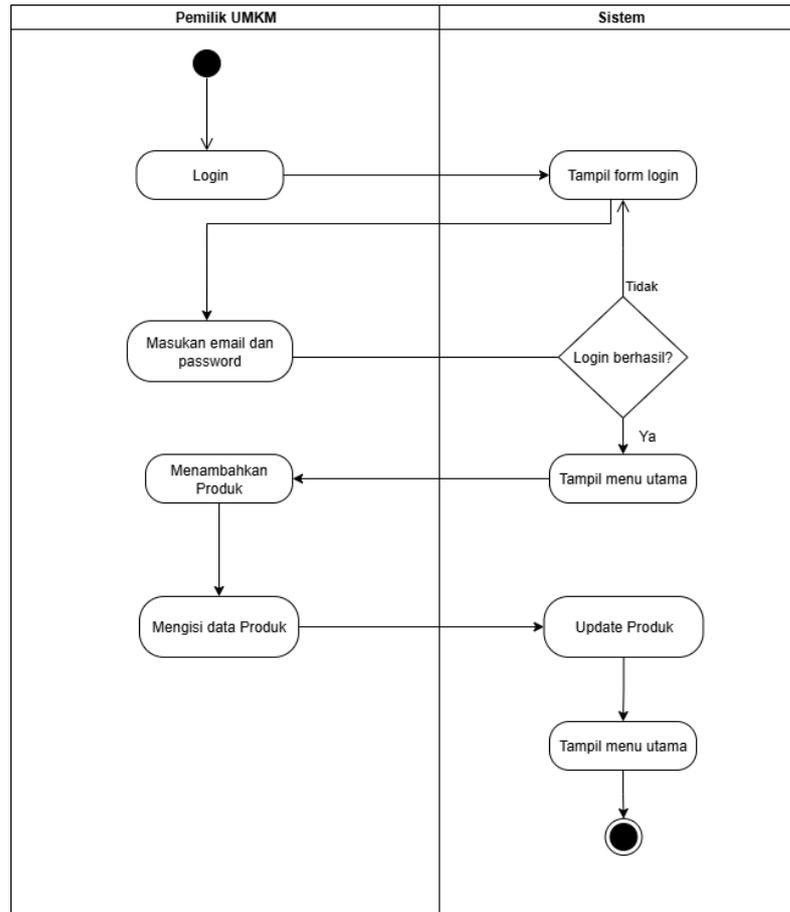
Pengguna memberikan data untuk registrasi kepada admin. Kemudian admin memasukkan data pengguna dan mengkonfirmasi serta meng-*update* data pengguna. Setelah itu pengguna dapat *login* untuk masuk halaman *website*.



Gambar 3. 8 Activity Diagram Proses Registrasi

2. *Activity Diagram* Proses Penambahan Produk

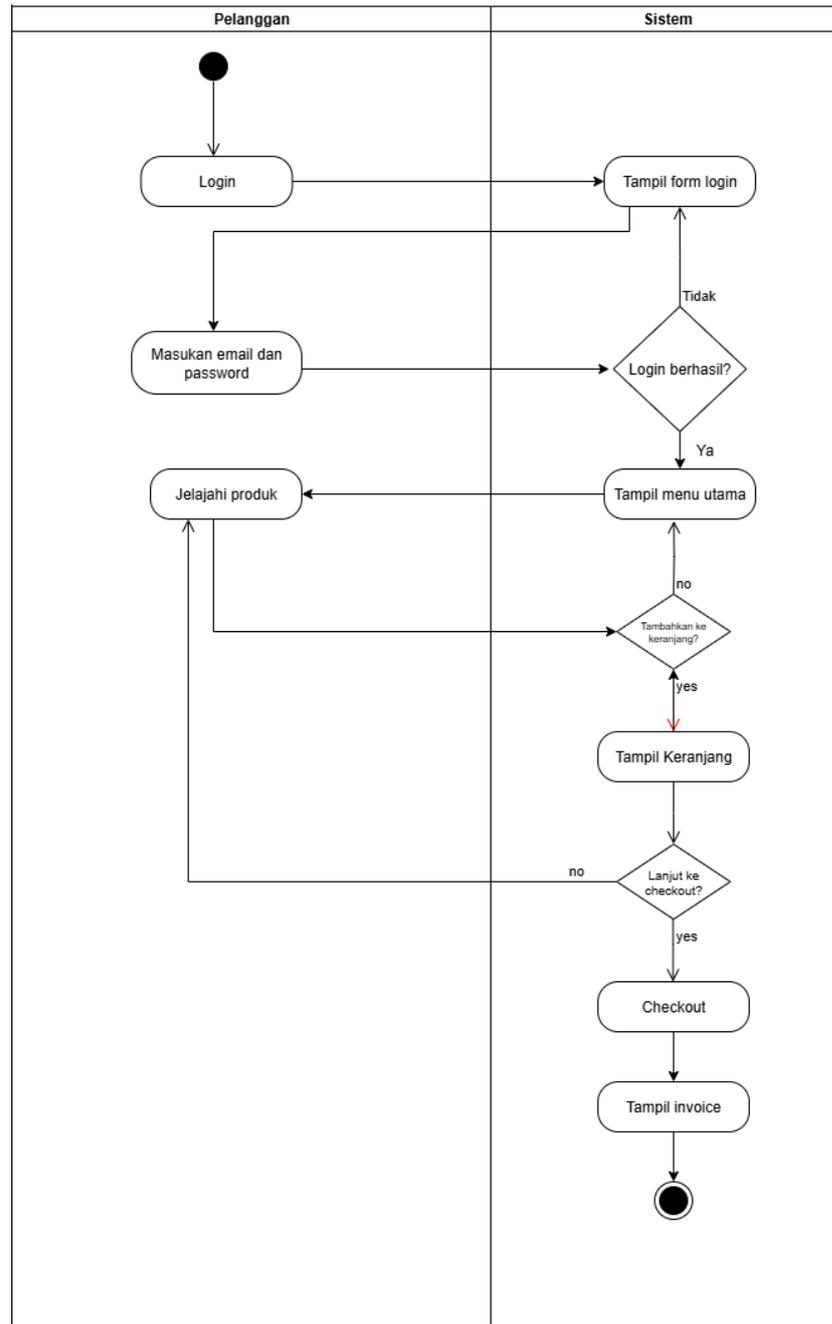
Aktifitas ini dilakukan Pemilik UMKM mulai dari *login* lalu menambahkan produk yang akan dijual.



Gambar 3.9 Activity Diagram Proses Penambahan Produk

3. Activity Diagram Proses Checkout

Dimulai dari pelanggan mulai *login* lalu menambahkan produk ke keranjang kemudian *Checkout* dan ditampilkan *invoice* untuk dikirimkan ke Pemilik UMKM.

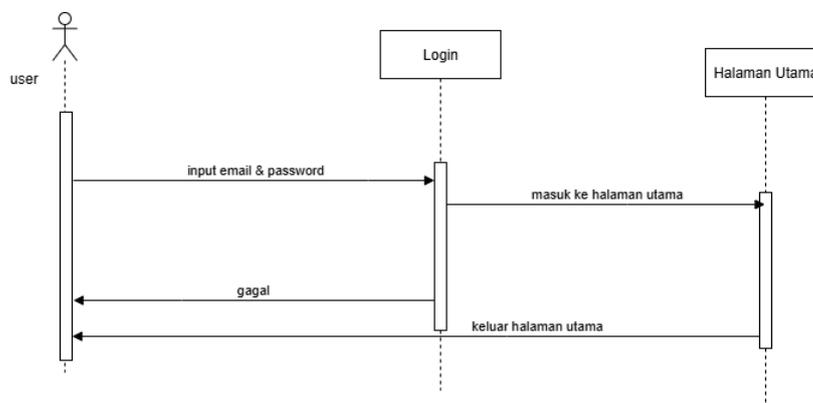


Gambar 3. 10 Activity Diagram Proses Checkout

3.2.2.3 Sequence Diagram

Sequence diagram (diagram urutan) adalah salah satu diagram perilaku (*behavioral diagram*) dalam *Unified Modeling Language (UML)* yang memodelkan interaksi dinamis antar objek/komponen sistem secara kronologis.

Fokus utamanya adalah urutan pengiriman pesan (*messages*) bukan struktur data sehingga diagram ini menunjukkan bagaimana kolaborasi objek mewujudkan suatu skenario fungsional.



Gambar 3. 11 Sequence Diagram

Diagram urutan (*Sequence diagram*) yang digunakan sebagai bagian dari perancangan proses dalam sistem *e-commerce* UMKM, khususnya untuk proses *login* pengguna. Diagram ini menggambarkan interaksi antar objek secara kronologis dalam menjalankan proses autentikasi pengguna ke dalam sistem.

Terdapat tiga objek utama dalam diagram ini, yaitu:

1. Pengguna (*User*)
2. Sistem *Login*
3. Halaman Utama

Alur proses dimulai saat pengguna menginput *email* dan *password* ke dalam *form login*. Informasi tersebut dikirim ke objek *Login* untuk proses verifikasi. Sistem kemudian memvalidasi *input-an* tersebut:

1. Jika *email* dan *password* tidak valid, maka sistem akan memberikan respon gagal, dan pengguna tidak dapat mengakses halaman utama.

2. Jika *email* dan *password* valid, sistem akan memberikan respon masuk ke halaman utama, dan pengguna diarahkan ke objek Halaman Utama.

3.2.3 Rancangan Keluaran (*Output*)

Perancangan keluaran (*Output*) dalam bentuk *invoice* pesanan pada sistem *e-commerce* UMKM. Elemen-elemen penting yang disajikan pada *invoice* meliputi: judul "*INVOICE PESANAN*", nama *website* (UMKM *Online Shop*), serta informasi rinci seperti *ID* Pesanan, Nama Pelanggan, Tanggal, dan Status Pemesanan.

Di bawah informasi tersebut, terdapat tabel yang menyajikan rincian pesanan, mencakup kolom Nomor, Nama Produk, Harga, Jumlah, dan Subtotal, yang memungkinkan pelanggan mengetahui secara jelas rincian transaksi yang dilakukan. Di bagian bawah tabel, terdapat baris penjumlahan dengan label *TOTAL* yang menunjukkan jumlah keseluruhan pembayaran. Terdapat juga tautan "*Kembali ke Daftar Pesanan*" untuk memudahkan navigasi pengguna kembali ke halaman sebelumnya.

INVOICE PESANAN				
UMKM Online Shop				
ID Pesanan:				
Pelanggan:				
Tanggal:				
Status:				
No	Nama Produk	Harga	Jumlah	Subtotal
1		Rp. -		Rp. -
2		Rp. -		Rp. -
Total:				Rp.
Kembali ke daftar pesanan				

Gambar 3. 12 Rancangan Keluaran (*Output*)

3.2.4 Rancangan *Interface (User Interface)*

Rancangan antarmuka dibuat berbasis prinsip usability (kemudahan dipahami, konsistensi, *feedback* jelas) dan *mobile-first responsive design*.

Wireframe (deskriptif):

1. Halaman Profil Pengguna

Halaman profil pengguna dirancang sebagai antarmuka untuk memudahkan pengguna dalam mengelola data profil. Pada bagian atas terdapat judul halaman dan ikon foto profil yang dapat diganti sesuai kebutuhan. Formulir yang tersedia terdiri atas kolom Nama Lengkap, *Email*, *Password* Baru, unggah foto profil, serta *Role* yang menunjukkan peran pengguna dalam sistem, seperti admin, penjual, atau pembeli. Pada bagian bawah terdapat tombol “Simpan Perubahan” yang berfungsi menyimpan pembaruan ke dalam basis data. Secara keseluruhan, halaman ini berfungsi sebagai pusat pengelolaan akun, memungkinkan pengguna memperbarui informasi pribadi, menjaga keamanan melalui pembaruan kata sandi, serta memastikan validitas data dalam mendukung kelancaran transaksi *online*.

The image shows a user profile page titled "Akun Saya". It features a circular profile picture placeholder. Below the profile picture, there are several input fields for user information: "Nama Lengkap", "Email", "Password Baru", "Foto Profil" (with a "Choose file" button), and "Role". At the bottom of the form is a "Simpan Perubahan" button.

Gambar 3. 13 Halaman Profil Pengguna

2. Halaman Katalog Produk

The image shows a product catalog page titled "Katalog Produk UMKM". It features a "Tambah Produk" button at the top center. Below the button, there are three rounded rectangular boxes representing product items, labeled "Produk 1", "Produk 2", and "Produk 3".

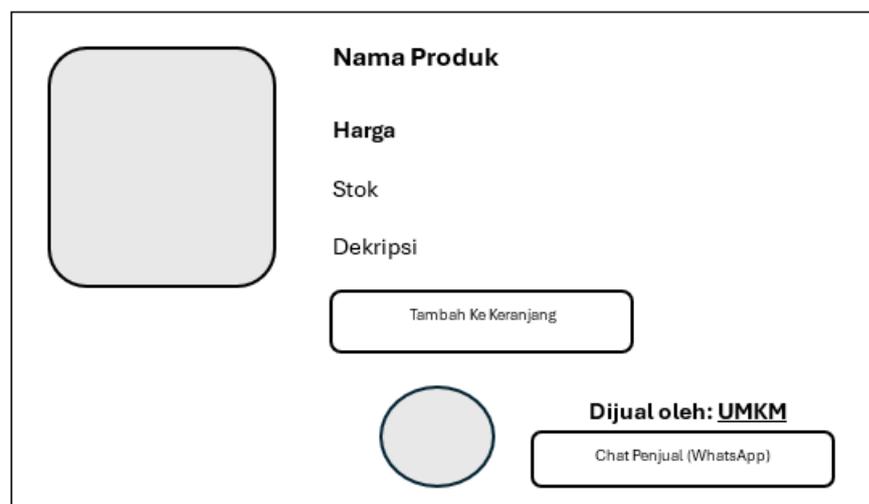
Gambar 3. 14 Halaman Katalog Produk

Pada halaman Katalog Produk terdapat judul “Katalog Produk UMKM” terletak di bagian paling atas tengah halaman yang menjadi penanda bahwa *user* telah berada pada halaman utama katalog, *user* yang dimaksud di sini adalah

Pemilik UMKM karna hanya *user* tersebut yang dapat melihat halaman ini. Dalam halaman ini terdapat fitur “Tambah Produk” yang digunakan *user* untuk menambahkan produk yang akan ditampilkan pada halaman katalog untuk dijual. Dibawah fitur “Tambah Produk” terdapat produk yang telah ditambahkan ke halaman ini.

3. Halaman Detail Produk

Halaman Detail Produk dirancang untuk menampilkan informasi lengkap mengenai suatu barang yang dijual. Pada bagian kiri terdapat gambar produk, sementara di sisi kanan ditampilkan nama produk, harga, stok, dan deskripsi secara ringkas. Pengguna dapat menambahkan barang yang dipilih ke keranjang melalui tombol “Tambah ke Keranjang” yang tersedia. Selain itu, halaman ini juga menampilkan informasi penjual (UMKM) beserta tombol “*Chat Penjual*” yang terhubung langsung ke *WhatsApp* untuk mempermudah komunikasi. Dengan rancangan ini, halaman detail produk berfungsi sebagai sarana utama bagi pengguna dalam memperoleh informasi, melakukan interaksi dengan penjual, serta memulai proses pembelian dengan lebih praktis.



Gambar 3. 15 Detail Produk

4. Halaman Keranjang

Halaman ini menunjukkan kumpulan produk yang kemungkinan akan dibeli oleh pelanggan. Terdapat kolom “Nama Produk” yang berisikan produk yang dimasukkan ke dalam keranjang belanja, lalu kolom “Harga” yang menunjukkan harga per-item produk, kolom “Jumlah” ialah jumlah item produk yang dimasukkan ke keranjang belanja, tabel “Subtotal” yaitu jumlah harga per-produk yang akan dibayar, dan selanjutnya kolom “Aksi” ini digunakan untuk menghapus produk apabila pelanggan tidak ingin membeli produk yang telah dimasukkan ke dalam keranjang. Harga total belanja terdapat pada sisi terbawah tabel. Jika pelanggan sudah yakin dan ingin membeli produk, maka pelanggan dapat memilih opsi “*Checkout Sekarang*” yang ada pada bawah tabel. Dan jika pelanggan ingin menambahkan produk ke dalam keranjang maka pelanggan dapat memilih opsi “Kembali ke Katalog Produk”, setelah memilih opsi tersebut pelanggan akan kembali ke halaman katalog yang dikhususkan untuk pelanggan.

Keranjang Belanja Saya					
No	Nama Produk	Harga	Jumlah	Subtotal	Aksi
1		Rp. -		Rp. -	Hapus
2		Rp. -		Rp. -	Hapus
				Total: Rp.	
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block; margin: 10px 0;">Checkout Sekarang</div> <small>Kembali ke Katalog Produk</small>					

Gambar 3. 16 Halaman Keranjang

5. Halaman *Invoice* Pesanan

Halaman “*Invoice* Pesanan” dari sistem *e-commerce* *UMKM Online Shop* yang berfungsi sebagai bukti transaksi pembelian. Halaman ini memuat informasi penting seperti *ID* Pesanan, Nama Pelanggan, dan Tanggal, serta menampilkan detail pesanan dalam tabel berisi Nama Produk, Harga, Jumlah, dan Subtotal. Di bagian bawah, terdapat kolom Total untuk menunjukkan jumlah pembayaran akhir. Tersedia juga tombol “Cetak *Invoice*” untuk mencetak dokumen, serta tautan “Kembali ke daftar pesanan” untuk navigasi.

INVOICE PESANAN
UMKM Online Shop

ID Pesanan:
Pelanggan:
Tanggal:
Status:

No	Nama Produk	Harga	Jumlah	Subtotal
1		Rp. -		Rp. -
2		Rp. -		Rp. -
Total:				Rp.

[Kembali ke daftar pesanan](#)

Gambar 3. 17 Halaman *Checkout*

6. *Dashboard* Admin:

Pada *dashoard* admin terdapat halaman “Daftar *User*” dalam sebuah sistem manajemen, ini berfungsi untuk menampilkan dan mengelola data pengguna.

Daftar User				
<input type="button" value="Tambah User"/>		<input type="button" value="Lihat Produk"/>		
No	Nama	Email	Role	Aksi
1				
2				

Gambar 3. 18 Dashboard Admin

Di bagian atas terdapat tombol “Tambah *User*” yang memungkinkan admin untuk menambahkan pengguna baru ke sistem. Di bawahnya, terdapat tabel yang berisi kolom Nomor, Nama, *Email*, *Role*, dan Aksi. Tabel ini digunakan untuk menampilkan daftar pengguna yang terdaftar dalam sistem, termasuk informasi peran masing-masing menyediakan opsi aksi seperti edit atau hapus. Tombol “Lihat Produk” berguna untuk mengontrol produk yang dijual oleh UMKM. Admin memiliki akses untuk mengedit dan menghapus produk.

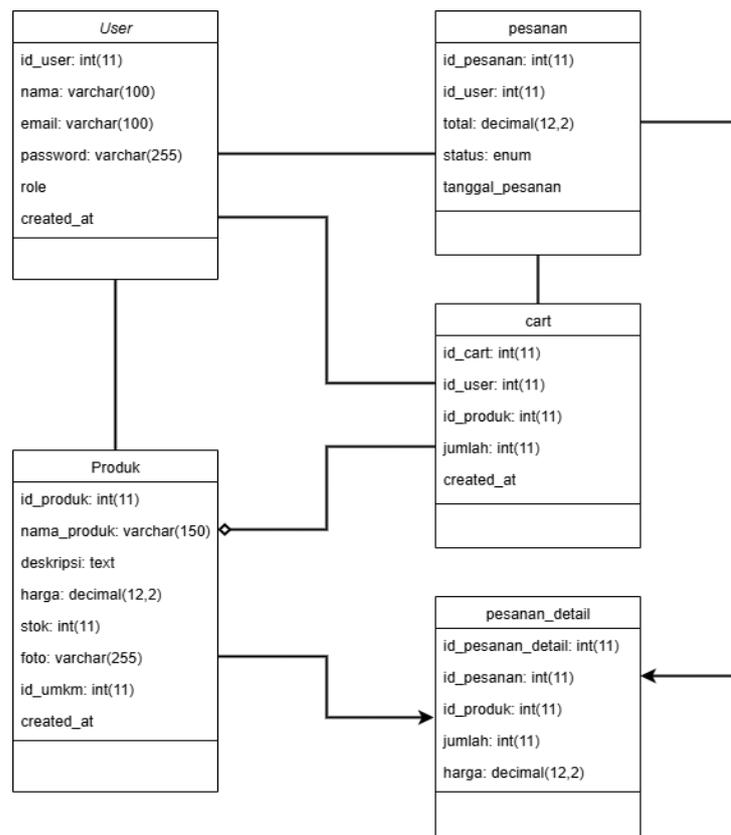
3.3 Perancangan Basis Data

Perancangan basis data merupakan tahap perencanaan struktur penyimpanan data yang digunakan oleh sistem agar pengelolaan informasi dapat dilakukan secara terorganisir, konsisten, dan efisien. Pada *website* penjualan *online* produk UMKM berbasis *CodeIgniter*, basis data dirancang untuk mendukung seluruh proses bisnis, mulai dari pengelolaan produk, transaksi, pelanggan, hingga manajemen pengguna.

3.3.1 Class Diagram

Class diagram merupakan salah satu bagian dari perancangan Basis Data yang menggambarkan struktur kelas dalam sistem beserta atribut, metode, dan relasi antar kelas.

Diagram Entity Relationship Diagram (ERD) dari sistem penjualan online produk UMKM yang terdiri dari enam entitas utama, yaitu *User*, *Produk*, *Cart*, *Pesanan*, *Pesanan_Detail*, dan relasi antar entitas tersebut.



Gambar 3. 19 Class Diagram

Entitas *User* menyimpan data pengguna yang terdiri dari atribut yakni *id_user*, *nama*, *email*, *password*, *alamat*, dan *no_hp*. Setiap user dapat melakukan transaksi dan memiliki lebih dari satu pesanan. Entitas *Produk* menyimpan informasi produk UMKM yaitu *id_produk*, *nama_produk*, *deskripsi*, *harga*, *stok*,

dan gambar. Produk yang tersedia dapat dimasukkan ke dalam keranjang (*cart*) oleh *user*.

Entitas *Cart* berfungsi sebagai tempat sementara untuk menyimpan produk yang akan dibeli. Atributnya meliputi *id_cart*, *id_user*, *id_produk*, jumlah, dan subtotal. Relasi antara *User*, *Cart*, dan Produk menggambarkan bahwa *user* dapat memasukkan banyak produk ke dalam *cart*.

Setelah *user* melakukan proses *checkout*, data akan berpindah ke entitas Pesanan, yang mencatat transaksi dengan atribut *id_pesanan*, *id_user*, *tgl_pesanan*, total, dan status. Detail dari setiap pesanan disimpan dalam entitas *Pesanan_Detail*, yang mencakup *id_pesanan_detail*, *id_pesanan*, *id_produk*, jumlah, dan subtotal.

Hubungan antar entitas bersifat relasional yaitu: satu *user* dapat membuat banyak pesanan, satu pesanan memiliki banyak detail pesanan, satu produk bisa muncul di banyak *cart* dan detail pesanan, serta satu user dapat memiliki banyak item dalam *cart*.

3.3.2 Tabel *users*

<i>Field</i>	<i>Type</i>	<i>Size</i>	Keterangan
<i>id_user</i>	<i>int</i>	11	<i>Primary key</i>
<i>nama</i>	<i>varchar</i>	100	Menyimpan nama lengkap pengguna
<i>email</i>	<i>varchar</i>	100	Menyimpan alamat <i>email</i> pengguna, bisa digunakan untuk <i>login</i>
<i>password</i>	<i>varchar</i>	255	Menyimpan hash dari <i>password</i> pengguna
<i>role</i>	<i>enum</i>		Menyimpan peran pengguna: 'admin', 'umkm', 'pelanggan'
<i>created_at</i>	<i>timestamp</i>		Menyimpan waktu saat akun pengguna dibuat

Tabel 3. 1 User

Tabel ini berfungsi untuk menyimpan data pengguna (*user*) dari sistem, baik admin maupun pelanggan. Tabel ini digunakan dalam sistem *login* dan manajemen hak akses pengguna.

3.3.3 Tabel Produk

Tabel ini digunakan untuk menyimpan informasi detail mengenai produk yang dijual oleh UMKM dalam sistem. Setiap *record* mewakili satu produk.

<i>Field</i>	<i>Type</i>	<i>Syze</i>	Keterangan
id_produk	<i>int</i>	11	<i>Primary key</i>
nama_produk	<i>varchar</i>	150	Menyimpan nama produk
deskripsi	<i>text</i>	100	Menyimpan deskripsi produk
harga	<i>decimal</i>	12,2	Menyimpan harga produk dengan dua angka desimal
stok	<i>int</i>	11	Menyimpan <i>UML</i> stok produk yang tersedia
foto	<i>varchar</i>	255	Menyimpan nama file atau path gambar produk
id_umkm	<i>int</i>	11	<i>Foreign Key</i> mengacu ke tabel UMKM (pemilik produk)
created_at	<i>timestamp</i>		Mencatat waktu saat data produk ditambahkan ke sistem

Tabel 3. 2 Produk

3.3.4 Tabel Pesanan Detail

Tabel ini berfungsi sebagai tabel relasi antara pesanan dan produk. Setiap baris merepresentasikan satu jenis produk dalam suatu pesanan tertentu. Tabel ini memungkinkan satu pesanan berisi banyak produk (*many-to-many relationship* antara pesanan dan produk).

<i>Field</i>	<i>Type</i>	<i>Syze</i>	Keterangan
id_pesanan_detail	<i>int</i>	11	<i>Primary key</i>
id_pesanan	<i>int</i>	11	<i>Foreign Key</i> , merujuk ke tabel pesanan
id_produk	<i>int</i>	11	<i>Foreign Key</i> , merujuk ke tabel produk
Jumlah	<i>int</i>	11	Menyimpan jumlah item produk yang dipesan
harga	<i>decimal</i>	12,2	Harga per unit produk pada saat pesanan dilakukan (bisa berbeda dari harga kini)

Tabel 3. 3 Pesanan detail

3.3.5 Tabel Pesanan

<i>Field</i>	<i>Type</i>	<i>Syze</i>	Keterangan
id_pesanan	<i>int</i>	11	<i>Primary key</i>
id_user	<i>int</i>	11	<i>Foreign Key</i> , merujuk ke tabel <i>users</i> sebagai pihak yang melakukan pemesanan
total	<i>decimal</i>	12,2	Total biaya dari seluruh item dalam pesanan (termasuk kuantitas dan harga)
status	<i>enum</i>		Status pemesanan: 'pending', 'diproses', 'selesai', 'dibatalkan'
tanggal_pesanan	<i>timestamp</i>		Waktu saat pesanan dibuat

Tabel 3. 4 Pesanan

Tabel ini menyimpan informasi utama dari suatu transaksi pemesanan yang dilakukan oleh pengguna. Tabel ini akan terhubung ke tabel *users* (melalui *id_user*) dan ke tabel *pesanan_detail* (melalui *id_pesanan*).

3.3.6 Tabel *Cart*

<i>Field</i>	<i>Type</i>	<i>Size</i>	Keterangan
<i>id_Cart</i>	int	11	<i>Primary key</i>
<i>id_user</i>	int	11	<i>Foreign Key</i> , merujuk ke pengguna yang menambahkan produk ke keranjang
<i>id_produk</i>	int	11	<i>Foreign Key</i> , merujuk ke produk yang ditambahkan ke keranjang
jumlah	int	11	Jumlah item produk yang ditambahkan
<i>created_at</i>	timestamp		Menyimpan waktu kapan item ditambahkan ke dalam keranjang

Tabel 3. 5 *Cart*

Tabel ini berfungsi sebagai tempat penyimpanan sementara produk-produk yang dipilih oleh pengguna sebelum melakukan *checkout* dan menyelesaikan pemesanan.