

Implementasi E-Commerce Berbasis Web Pada Saponik Farm

Agus Ranaputra^{1*}, Alfin Albino², Erwin Nurrudin Syahputra³, Nunung Hidayatun⁴

^{1,2,3,4}Universitas Bina Sarana Informatika
Jl. Raya Kaliabang No.8, Perwira, Kota Bekasi, Jawa Barat, Indonesia

e-mail korespondensi: agusranaputra@gmail.com

Submit: 22-10-2025 | Revisi: 07-11-2025 | Terima: 10-11-2025 | Terbit online: 20-11-2025

Abstrak - Aplikasi *E-Commerce* berbasis web mempermudah para pengusaha dalam memasarkan produk mereka serta membantu pelanggan mendapatkan informasi tentang barang yang ditawarkan oleh penjual. Saat ini, sistem informasi yang tersedia secara daring dimanfaatkan sebagai alat untuk meningkatkan penyampaian informasi. Saponik Farm adalah sebuah usaha yang bergerak di bidang penjualan sayuran hidroponik. Selama ini, kendala dalam pemasaran produk menjadi penghambat bagi penjual untuk meningkatkan pendapatan dari penjualan, ditambah dengan sistem pemrosesan penjualan yang belum efektif, serta tantangan dalam menjangkau pelanggan baru. Penelitian ini bertujuan merancang dan mengimplementasikan sistem *E-Commerce* berbasis web untuk mempermudah transaksi, memperluas jangkauan calon pelanggan dan pengelolaan data penjualan. Metode penelitian yang digunakan adalah metode pengumpulan data yang terdiri dari observasi, wawancara dan studi pustaka serta metode pengembangannya menggunakan model *Waterfall* dengan tahapan analisis, perancangan, implementasi, dan pengujian. Sistem dibangun menggunakan *framework CodeIgniter* dengan arsitektur *Model-View-Controller* (MVC) dan basis data *MySQL*. Hasil implementasi menunjukkan sistem mampu mendukung transaksi online dan memudahkan pengelolaan penjualan secara digital.

Kata Kunci : E-Commerce, CodeIgniter, Saponik Farm, Waterfall

Abstracts - *Web-based E-Commerce applications facilitate entrepreneurs in marketing their products and help customers obtain information about the goods offered by sellers. Currently, online information systems are used as a tool to improve information delivery. Saponik Farm is a business engaged in the sale of hydroponic vegetables. So far, obstacles in product marketing have been a barrier for sellers to increase revenue from sales, coupled with an ineffective sales processing system, as well as challenges in reaching new customers. This study aims to design and implement a web-based E-Commerce system to facilitate transactions, expand the reach of potential customers and manage sales data. The research method used is a data collection method consisting of observation, interviews and literature studies. The development method uses the Waterfall model with the stages of analysis, design, implementation, and testing. The system was built using the CodeIgniter framework with a Model-View-Controller (MVC) architecture and a MySQL database. The implementation results show that the system is able to support online transactions and facilitate digital sales management.*

Keywords : E-Commerce, CodeIgniter, Saponik Farm, Waterfall

1. Pendahuluan

Pada era digital yang semakin maju, perkembangan teknologi informasi telah membawa perubahan signifikan terhadap berbagai aspek kehidupan manusia. Teknologi tidak hanya berperan dalam mempercepat proses komunikasi dan pertukaran data, tetapi juga memberikan dampak besar pada bidang transportasi, pendidikan, kesehatan, serta kegiatan ekonomi masyarakat [1]. Salah satu wujud nyata dari kemajuan teknologi tersebut adalah munculnya sistem perdagangan elektronik atau *electronic commerce (E-Commerce)* yang memungkinkan terjadinya transaksi jual beli secara daring melalui jaringan internet. *E-Commerce* menjadi sarana utama bagi perusahaan maupun pelaku usaha kecil dan menengah dalam memperluas jangkauan pasar tanpa batas ruang dan waktu. Model perdagangan ini menjadi solusi efisien bagi pelaku bisnis dalam meningkatkan daya saing dan mempercepat proses transaksi di tengah perkembangan teknologi digital yang semakin pesat [2].

Beberapa penelitian sebelumnya telah menunjukkan bahwa penerapan sistem *E-Commerce* berpengaruh positif terhadap efisiensi bisnis dan kepuasan pelanggan. Penelitian oleh Jannah dan Nugraha [2] menjelaskan bahwa *E-Commerce* mampu memberikan fleksibilitas dalam proses jual beli, mempercepat transaksi, dan meningkatkan komunikasi antara penjual dan pembeli. Selanjutnya, Wiawan et al. [3] menegaskan bahwa pemanfaatan teknologi berbasis web pada dunia bisnis membantu penyebaran informasi secara cepat dan akurat,



sehingga mendukung peningkatan pelayanan kepada pelanggan. Selain itu, Ahadiyah [4] menyoroti bahwa perkembangan teknologi informasi menjadi katalis bagi para pengusaha dalam mengoptimalkan kegiatan operasional dan memperluas jangkauan pasar melalui *platform* digital. Berdasarkan berbagai hasil penelitian tersebut, dapat disimpulkan bahwa sistem penjualan berbasis web berperan penting dalam memperbaiki sistem manajemen produk serta memberikan kemudahan dalam transaksi dan pencatatan penjualan.

Namun demikian, hasil observasi menunjukkan bahwa masih terdapat kesenjangan penerapan teknologi informasi pada sektor usaha mikro, seperti yang terjadi di SAponik Farm - sebuah usaha mikro di bidang pertanian hidroponik. Proses penjualan di SAponik Farm saat ini masih dilakukan secara konvensional melalui media sosial, *chat* pribadi, maupun penjualan langsung ke konsumen. Metode ini memiliki berbagai keterbatasan, seperti cakupan pasar yang sempit, proses transaksi yang tidak efisien, serta dokumentasi penjualan yang tidak terkomputerisasi. Kondisi tersebut menimbulkan *research gap* berupa kebutuhan akan sistem informasi penjualan berbasis web yang mampu meningkatkan efisiensi, memperluas jangkauan pasar, serta mempermudah manajemen stok dan transaksi penjualan. Oleh karena itu, penelitian ini berfokus pada perancangan dan implementasi sistem *E-Commerce* berbasis web pada SAponik Farm sebagai solusi untuk mengoptimalkan proses penjualan dan meningkatkan daya saing usaha hidroponik di era digital.

2. Metode Penelitian

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *Waterfall*. Metode ini merupakan pendekatan sistematis dan terstruktur yang dilakukan secara bertahap mulai dari tahap analisis hingga pemeliharaan sistem [5]. Dengan menerapkan model *Waterfall*, penulis dapat membangun sistem *E-Commerce* berbasis web secara terencana dan berurutan, sehingga setiap tahap pengembangan dapat dievaluasi dengan baik sebelum melangkah ke tahap berikutnya.

a. Analisis Kebutuhan (*Requirements Analysis*)

Tahapan tahap pertama adalah analisis kebutuhan sistem, yang dilakukan untuk memahami proses bisnis di SAponik Farm serta mengidentifikasi permasalahan yang ada pada sistem penjualan konvensional. Pengumpulan data dilakukan melalui observasi langsung, wawancara dengan pemilik usaha, serta studi literatur yang relevan. Dari hasil analisis tersebut, diperoleh kebutuhan fungsional sistem seperti pengelolaan produk, transaksi pemesanan, dan pencatatan laporan penjualan. Alur proses bisnis yang telah dianalisis kemudian dimodelkan menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* agar hubungan antar aktor dan proses sistem dapat digambarkan secara visual dan logis. Tahapan ini menjadi dasar penting dalam merancang struktur sistem yang sesuai dengan kebutuhan pengguna.

b. Perancangan (*Design*)

Tahap kedua adalah perancangan sistem, yang bertujuan untuk mengubah hasil analisis kebutuhan menjadi model yang dapat diimplementasikan. Perancangan dilakukan dalam dua aspek utama, yaitu perancangan basis data dan perancangan antarmuka pengguna. Model basis data yang dibuat mencakup pembuatan model data seperti ERD, LRS, dan *Class Diagram* yang menggambarkan hubungan antar-entitas serta atribut yang digunakan, serta rancangan antarmuka pengguna yang menggambarkan tampilan sistem [6]. Sedangkan rancangan antarmuka pengguna (*user interface design*) dibuat untuk memvisualisasikan tampilan halaman web, seperti halaman beranda, katalog produk, form pemesanan, dan halaman admin. Desain antarmuka memperhatikan prinsip kemudahan penggunaan (*usability*) dan konsistensi agar sistem mudah dipahami oleh pengguna.

c. Implementasi/Pengkodean (*Implementation*)

Tahap ketiga, implementasi, dilakukan dengan menerjemahkan hasil rancangan ke dalam kode program menggunakan bahasa *PHP*, *HTML*, *CSS*, dan basis data *MySQL*, *MySQL* termasuk salah satu perangkat lunak sistem untuk mengelola *database SQL* [7]. *Framework CodeIgniter* digunakan untuk mempercepat pengembangan, sedangkan *XAMPP Apache* berfungsi sebagai *server* lokal untuk pengujian dan debugging. Seluruh komponen diintegrasikan sehingga membentuk sistem *E-Commerce* berbasis web yang fungsional dan terhubung antarhalaman.

d. Pengujian (*Testing*)

Tahap keempat, pengujian, dilakukan menggunakan metode *Blackbox Testing* yang bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan pengguna. Pengujian dilakukan menggunakan metode *Blackbox Testing*, yaitu pengujian yang berfokus pada keluaran (*output*) berdasarkan masukan (*input*) tanpa memperhatikan struktur internal program. Setiap fungsi utama seperti login, pengelolaan produk, transaksi pemesanan, dan pencetakan laporan diuji secara terpisah untuk memastikan tidak ada kesalahan logika maupun tampilan. Pengujian dilakukan pada dua peran pengguna, yaitu *admin* (pemilik usaha) dan *user* (pelanggan), untuk memastikan sistem berfungsi baik dari sisi operasional maupun layanan pengguna.

e. Pemeliharaan (*Maintenance*)

Tahap terakhir adalah pemeliharaan, yaitu perawatan dan pengembangan sistem setelah diimplementasikan agar tetap sesuai dengan kebutuhan pengguna dan dapat menyesuaikan dengan perubahan di masa mendatang

[8]. Tahapan ini juga mencakup evaluasi terhadap kinerja sistem berdasarkan masukan pengguna untuk menjaga agar aplikasi tetap relevan dan efektif dalam mendukung kegiatan penjualan SAponik Farm.

3. Hasil dan Pembahasan

3.1 Analisis Kebutuhan

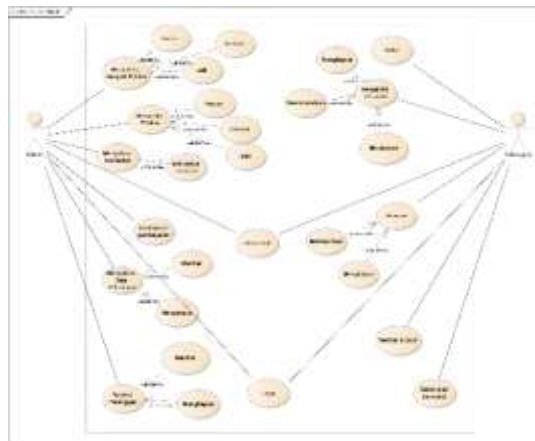
Aplikasi *E-Commerce* SAponik Farm dikembangkan untuk memenuhi kebutuhan digitalisasi penjualan produk sayuran hidroponik dengan dua jenis pengguna utama, yaitu *admin* dan pelanggan. *Admin* bertanggung jawab dalam pengelolaan data produk, transaksi, pelanggan, serta laporan penjualan. Sedangkan pelanggan dapat melakukan pendaftaran, *login*, melihat katalog produk, melakukan pemesanan, memberikan ulasan, serta memantau status transaksi. Dari sisi sistem, aplikasi ini dirancang agar mampu melakukan verifikasi pengguna, menampilkan data produk secara dinamis, memproses transaksi dan pengurangan stok otomatis, menampilkan laporan penjualan, serta menghasilkan bukti transaksi. Seluruh fungsi diuji untuk memastikan keandalan sistem dalam mendukung proses bisnis SAponik Farm.

3.2 Desain Sistem

Desain sistem dibuat menggunakan pendekatan *Unified Modeling Language* (UML) merupakan bahasa visual yang digunakan untuk merancang, memodelkan, dan mendiskusikan sistem perangkat lunak melalui kombinasi teks dan diagram tambahan. UML berfungsi sebagai alat bantu pemodelan dan tidak terbatas pada satu metode tertentu, meskipun pendekatan berorientasi objek merupakan yang paling umum digunakan [9]. Dalam penelitian ini, beberapa jenis diagram UML digunakan untuk menggambarkan struktur dan perilaku sistem secara menyeluruh, yaitu:

a. Use Case Diagram

Menggambarkan hubungan antara aktor (*admin* dan pelanggan) dengan sistem dalam aktivitas utama seperti *login*, pengelolaan produk, pemesanan, dan pembuatan laporan, berfungsi sebagai sarana komunikasi antara pengguna dan pengembang, menggambarkan kebutuhan sistem secara singkat serta hubungan antara sistem dengan aktor eksternal. Diagram ini berperan penting dalam mendukung pendekatan pengembangan berbasis pengalaman pengguna [10].



Gambar 1. Use Case Diagram

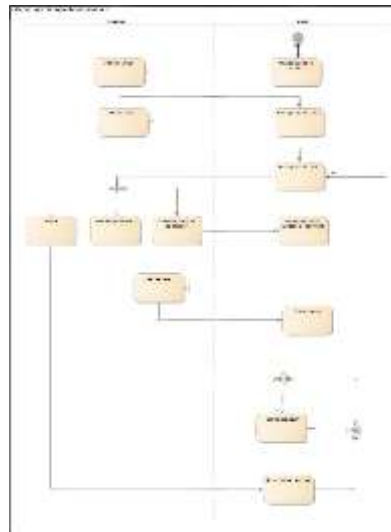
Use Case Diagram gambar 1, menampilkan sistem *E-Commerce* berbasis web pada SAponik Farm. Diagram ini terdiri dari dua aktor utama, yaitu *Admin* (Pemilik) dan *User* (Pelanggan). *Admin* memiliki hak akses untuk mengelola produk, memproses pesanan, memperbarui stok, dan mencetak laporan penjualan, sedangkan *User* dapat melakukan registrasi akun, melihat katalog produk, melakukan pemesanan, serta melakukan konfirmasi pembayaran. Setiap *use case* menggambarkan fungsi utama yang dapat dijalankan oleh masing-masing aktor. Hubungan antar-aktor dan sistem divisualisasikan melalui garis asosiasi yang menunjukkan interaksi atau komunikasi langsung. Dengan demikian, diagram ini memberikan gambaran awal mengenai batasan sistem, fungsi utama yang tersedia, serta bagaimana setiap aktor berinteraksi dengan sistem *E-Commerce* yang dikembangkan.

b. Activity Diagram

Digunakan untuk menunjukkan alur aktivitas dalam sistem, termasuk proses awal, keputusan yang mungkin terjadi, hingga akhir dari suatu aktivitas. Diagram ini memvisualisasikan urutan proses bisnis dan logika sistem secara jelas [11].

Activity Diagram pelanggan gambar 2 menjelaskan proses dimulai dari pengguna yang melakukan login ke dalam sistem. Setelah login berhasil, pengguna dapat melakukan aktivitas sesuai peran masing-masing, seperti admin yang mengelola data produk dan pesanan, serta pelanggan yang dapat melakukan pemesanan dan konfirmasi pembayaran. Setiap aktivitas ditandai dengan simbol action state, sedangkan proses pengambilan

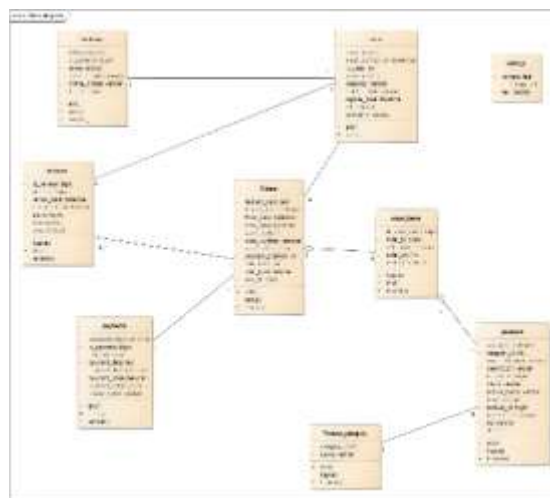
keputusan direpresentasikan oleh decision node. Dengan adanya Activity Diagram ini, pengembang dapat memahami urutan proses bisnis secara sistematis, memastikan bahwa setiap langkah berjalan sesuai kebutuhan pengguna.



Gambar 2. Activity Diagram pelanggan

c. Class Diagram

Digunakan untuk menunjukkan alur aktivitas dalam sistem, termasuk proses awal, keputusan yang mungkin terjadi, hingga akhir dari suatu aktivitas. Diagram ini memvisualisasikan urutan proses bisnis dan logika sistem secara jelas [12]. yang terdiri atas entitas pengguna, produk, kategori, pesanan, dan pembayaran. Relasi antar kelas mendukung modularitas program dan penerapan konsep *CRUD* (*Create, Read, Update, Delete*).

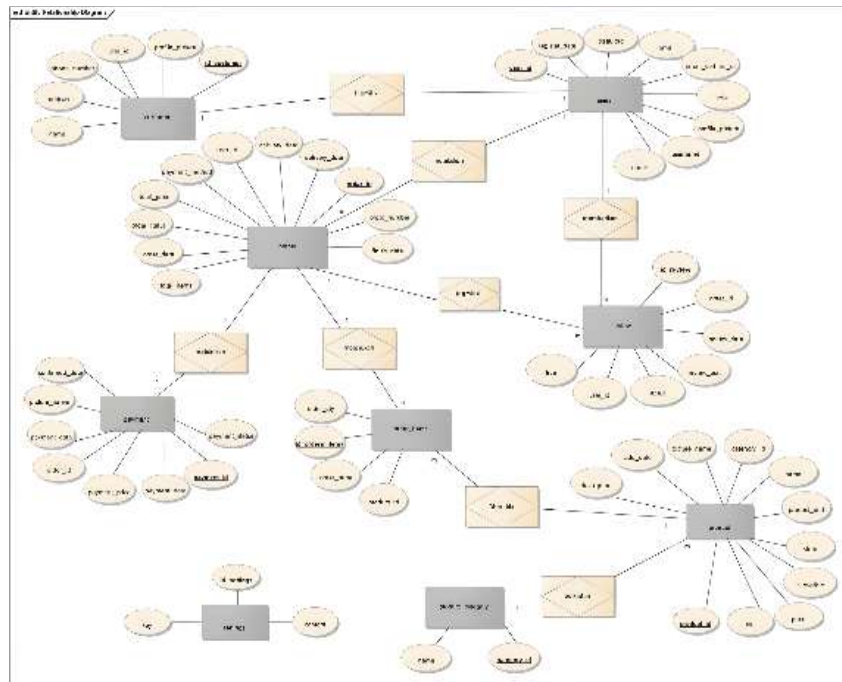


Gambar 3. Class Diagram

Class Diagram gambar 3, diagram ini terdiri atas beberapa kelas utama seperti Produk, Kategori, Pelanggan, Pesanan, dan Admin. Setiap kelas memiliki atribut dan metode sesuai fungsinya, misalnya kelas Produk memiliki atribut seperti *id_produk*, *nama_produk*, *harga*, dan *stok*, serta metode *CRUD* (*Create, Read, Update, Delete*). Hubungan antar kelas ditunjukkan melalui garis asosiasi dan relasi, misalnya antara kelas Pesanan dan Pelanggan terdapat relasi satu ke banyak, karena satu pelanggan dapat melakukan beberapa pesanan. Diagram ini membantu memahami struktur data dan hubungan antar komponen dalam sistem.

d. Entity Relationship Diagram (ERD)

Adalah skema konseptual yang digunakan untuk merancang basis data dengan menampilkan relasi antar entitas, atribut, serta keterkaitannya dalam suatu sistem [13]. ERD berfungsi sebagai alat bantu dalam penyusunan dan perancangan database, memberikan gambaran visual tentang struktur data, serta menjelaskan bagaimana data saling berhubungan di dalam sistem..

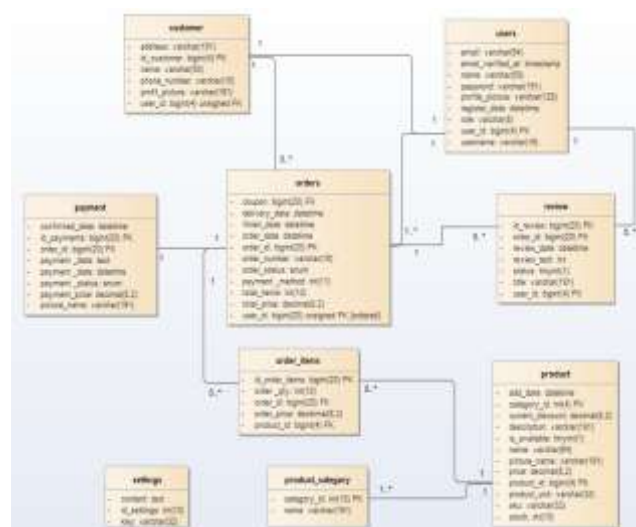


Gambar 4. Entity Relationship Diagram

Entity Relationship Diagram gambar 4, Pada sistem E-Commerce Saponik Farm, entitas utama yang terlibat meliputi produk, kategori, pelanggan, pesanan, dan detail_pesanan. Hubungan antar entitas digambarkan dengan kardinalitas satu ke banyak (1:N) atau banyak ke banyak (M:N), sesuai kebutuhan sistem. Sebagai contoh, satu pelanggan dapat memiliki banyak pesanan, tetapi satu pesanan hanya dimiliki oleh satu pelanggan. Dengan demikian, ERD berperan penting dalam merancang struktur basis data yang efisien dan konsisten agar sistem dapat berjalan dengan baik.

e. Logical Record Structure (LRS)

Merupakan hasil turunan dari Entity Relationship (ER) yang berisi seluruh atribut serta menunjukkan hubungan antar entitas dalam sistem basis data. Struktur ini menggambarkan jenis-jenis rekaman (record type) yang diilustrasikan dalam bentuk kotak persegi panjang dan masing-masing memiliki nama yang merepresentasikan entitas tertentu [14]. LRS digunakan untuk memvisualisasikan rancangan logis dari sistem data agar hubungan antar entitas dapat dipahami secara lebih terstruktur.



Gambar 5. Logical Record Structure

Logical Record Structure gambar 5, menggambarkan hubungan antara atribut utama seperti id_produk, id_pesanan, dan id_pelanggan yang digunakan untuk menjaga keutuhan relasi data antar tabel [15]. Melalui

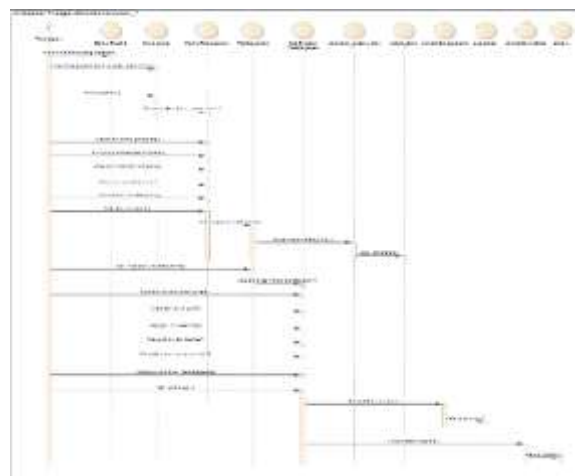
LRS, pengembang dapat memastikan bahwa rancangan basis data telah memenuhi kebutuhan sistem dan siap untuk diimplementasikan secara teknis.

f. *Sequence Diagram*

Menunjukkan alur pesan antar objek dalam suatu *use case* seiring berjalannya waktu. Diagram ini menjelaskan interaksi antar komponen sistem secara kronologis untuk menggambarkan proses bisnis yang terjadi [15], sehingga memperjelas interaksi antar objek dalam setiap skenario utama, seperti proses pemesanan dan konfirmasi pembayaran.

Sequence Diagram pada gambar 6 adalah skenario pemesanan produk, pengguna memulai interaksi dengan melakukan login, kemudian memilih produk yang diinginkan dan menambahkannya ke keranjang. Setelah itu, sistem menampilkan total harga dan menyediakan opsi pembayaran. Selanjutnya, pengguna melakukan konfirmasi pembayaran yang diteruskan ke admin untuk diverifikasi. Setelah pembayaran berhasil diverifikasi, sistem memperbarui status pesanan dan mengirimkan notifikasi kepada pengguna.

Seluruh diagram tersebut menjadi acuan dalam pembangunan sistem agar alur data, proses, dan interaksi pengguna berjalan konsisten dan efisien.



Gambar 6. *Sequence Diagram*

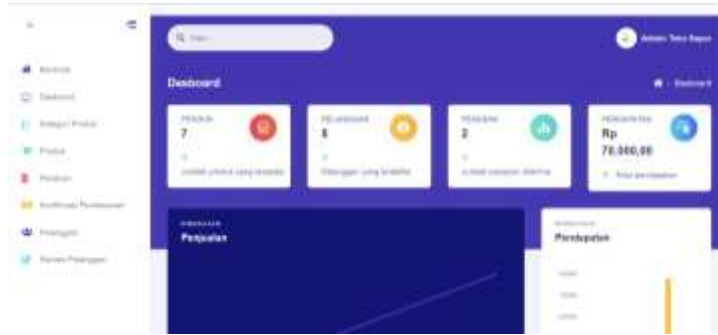
3.3 Implementasi Sistem

Aplikasi dikembangkan menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter dan basis data MySQL. Antarmuka pengguna (*user interface*) dibuat responsif agar dapat diakses melalui perangkat desktop maupun mobile.



Gambar 7. Halaman Utama Web

Gambar 7, merupakan tampilan halaman utama web aplikasi SAponk Farm yang mempunyai fitur utama meliputi pengelolaan katalog produk, keranjang belanja, *checkout*, konfirmasi pembayaran, dan laporan transaksi. Admin memiliki akses penuh untuk menambah, mengubah, dan menghapus data produk maupun transaksi.



Gambar 8. Halaman *Dashboard Admin*

Gambar 8, merupakan tampilan utama dan ringkasan informasi sistem dari seluruh aktivitas sistem. Pelanggan dapat melakukan pemesanan, pembayaran, serta memberikan ulasan terhadap produk.



Gambar 9. Halaman *Dashboard Pelanggan*

Gambar 9, merupakan tampilan utama dan ringkasan informasi sistem dari seluruh aktivitas sistem.

3.4 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *Blackbox Testing* untuk memastikan seluruh fungsi berjalan sesuai dengan kebutuhan pengguna. Hasil pengujian menunjukkan bahwa sistem dapat menjalankan seluruh proses utama dengan baik, termasuk Berhasilasi, transaksi pemesanan, penghitungan total belanja, serta pengelolaan stok produk

Tabel 1. Hasil *Blackbox Testing Login All Role*

NO	Scenario Pengujian	Test case	Keterangan
1	Login dengan <i>username</i> dan <i>password</i> Berhasil	Masukan <i>username</i> dan <i>password</i> yang sesuai - > klik <i>login</i>	Berhasil
2	Login dengan <i>username</i> dan <i>password</i> salah	Masukan <i>username</i> dan <i>password</i> yang salah-> klik <i>login</i>	Berhasil
3	Login dengan <i>username</i> benar dan <i>password</i> salah	Masukan <i>username</i> benar dan <i>password</i> yang salah-> klik <i>login</i>	Berhasil

Pada Tabel 1, Hasil *Blackbox Testing Login All Role* menunjukkan hasil pengujian fitur *Login* untuk seluruh peran pengguna (*admin* dan pelanggan). Dari hasil pengujian diperoleh bahwa proses autentikasi berjalan sesuai harapan. Sistem mampu mengenali kombinasi *username* dan *password* yang benar serta menolak yang salah dengan pesan kesalahan yang sesuai. Dengan demikian, keamanan akses pengguna dapat terjamin.

Tabel 2. Halaman *Dashboard Admin*

NO	Scenario Pengujian	Test case	Keterangan
1	Menampilkan data produk	Masuk ke menu produk	Berhasil
2	Menampilkan total pesanan	Masuk ke menu pesanan	Berhasil
3	Melihat data pelanggan	Menampilkan menu <i>dashboard</i> → lalu pilih pelanggan → klik nama pelanggan yang ingin dilihat	Berhasil
4	Menghapus data pelanggan	Menampilkan menu <i>dashboard</i> → lalu pilih pelanggan → pilih ikon sampah	Berhasil
5	Menambahkan data produk	Menampilkan menu <i>dashboard</i> → lalu pilih produk → klik tambah → lalu masukkan data produk yang ingin ditambahkan → Lalu klik tambah produk baru	Berhasil

NO	Scenario Pengujian	Test case	Keterangan
6	Menghapus data produk	Menampilkan menu <i>dashboard</i> → lalu pilih produk → klik ikon mata → lalu klik ikon sampah → Lalu klik hapus	Berhasil
7	Mengedit data produk	Menampilkan menu <i>dashboard</i> → lalu pilih produk → klik ikon <i>edit</i> → lalu masukkan data baru → Lalu klik simpan	Berhasil
8	Mengelola pesanan pelanggan	Menampilkan menu <i>dashboard</i> → lalu pilih pesanan → klik pesanan yang ingin dikelola → lalu ubah status pesanan → Lalu klik ok	Berhasil
9	Melihat dan menghapus <i>review</i> pelanggan	Menampilkan menu <i>dashboard</i> → lalu pilih <i>review</i> pelanggan → pilih data yang ingin dilihat → Apabila ingin menghapus → pilih hapus → Dan konfirmasi	Berhasil
10	Mengedit profil, nama, <i>username</i> , email, dan <i>password</i>	Menampilkan menu <i>dashboard</i> → pilih ikon <i>admin</i> → lalu masukkan data baru yang ingin di <i>edit</i> → lalu klik simpan	Berhasil

Pada Tabel 2, Halaman *Dashboard Admin* menggambarkan hasil pengujian pada halaman Dashboard Admin, yang berfungsi untuk mengelola data produk, pesanan, dan pelanggan. Berdasarkan hasil uji coba, seluruh skenario yang diujikan berhasil dijalankan dengan baik, termasuk menambah, menghapus, serta memperbarui data. Hal ini menunjukkan bahwa fungsi CRUD (*Create, Read, Update, Delete*) telah berjalan secara optimal.

Tabel 3. Pelanggan

NO	Scenario Pengujian	Test Case	Keterangan
1	Pelanggan dapat memasukan produk ke dalam keranjang	Memilih produk yang ingin di masukan keranjang, klik <i>add to cart</i>	Berhasil
2	Pelanggan dapat membuat pesanan	Masuk menu keranjang Buat pesanan → isi <i>form</i> → buat pesanan → konfirmasi pembayaran → pilih transfer bank dan masukan bukti pembayaran → konfirmasi	Berhasil
3	Membatalkan pesanan yang belum terbayar	Halaman <i>dashboard</i> pelanggan → order saya → pilih pesanan → pilih batalkan → konfirmasi	Berhasil
4	Membuat <i>review</i>	Halaman <i>dashboard</i> pelanggan → <i>review</i> → tulis <i>review</i> baru → isi <i>form</i> → pilih tulis <i>review</i>	Berhasil
5	Menghapus <i>review</i>	Halaman <i>dashboard</i> pelanggan → <i>review</i> → lalu pilih data <i>review</i> → lalu klik ikon hapus	Berhasil
6	Mengedit <i>profile</i> , akun dan email	Halaman <i>dashboard</i> → klik ikon <i>profile</i> → lalu <i>edit</i> data yang diinginkan → lalu klik ganti nama	Berhasil

Pada Tabel 3, Pelanggan menyajikan hasil pengujian pada halaman Pelanggan, di mana pengguna dapat menambahkan produk ke keranjang, melakukan pemesanan, serta memberikan ulasan terhadap produk. Dari hasil pengujian, sistem berhasil menampilkan data sesuai permintaan pengguna dan mampu menolak transaksi yang belum dikonfirmasi secara otomatis. Dengan demikian, fungsi transaksi dan manajemen pesanan telah berjalan secara stabil dan responsif.

3.5 Jadwal Implementasi

Tabel 4 menampilkan jadwal implementasi aplikasi *E-Commerce* berbasis web di *Saponik Farm*.

Tabel 4. Jadwal Implementasi

NO	KEGIATAN	BULAN 1			BULAN 2				BULAN 3					
		1	2	3	4	1	2	3	4	1	2	3	4	
1	Pengumpulan Data	■												
2	Analisa Sistem		■											
3	Perancangan Konseptual			■	■									
4	Pembuatan Aplikasi				■	■	■							
5	Pengujian (<i>Testing</i>)					■	■	■						
6	Penyeluhan Dan Training							■	■					
7	Test Sistem									■	■			
8	Peralihan Sistem											■	■	

Pada Tabel 4, Jadwal Implementasi memperlihatkan jadwal pelaksanaan setiap kegiatan. Pada bulan

pertama dilakukan pengumpulan data dan analisis sistem, diikuti tahap perancangan konseptual hingga pembuatan aplikasi di bulan kedua. Selanjutnya, bulan ketiga difokuskan pada tahap *testing*, pelatihan pengguna (*training*), serta evaluasi akhir untuk memastikan sistem siap dioperasikan secara penuh.

3.6 Spesifikasi *Hardware & Software*

Pengembangan aplikasi *E-Commerce* SAponik *Farm* membutuhkan perangkat keras dan perangkat lunak dengan spesifikasi minimum agar sistem dapat berjalan optimal.

- a) Perangkat Keras (*Hardware*) Aplikasi dijalankan pada komputer dengan prosesor minimal 2 GHz, memori RAM 8 GB, dan kapasitas penyimpanan 256 GB. Koneksi jaringan menggunakan LAN serta tampilan monitor beresolusi 1920×1080 piksel untuk mendukung antarmuka yang responsif.
- b) Perangkat Lunak (*Software*) Sistem dikembangkan dan diuji pada Windows 11 atau Linux Debian dengan web *server* Apache. Bahasa pemrograman yang digunakan adalah PHP versi 7 dengan basis data MySQL. Aplikasi diakses melalui peramban web seperti *Mozilla Firefox*, *Google Chrome*, atau peramban sejenis.

4. Kesimpulan

Implementasi sistem *E-Commerce* merupakan langkah strategis untuk memperluas penjualan SAponik *Farm* merupakan langkah strategis dalam meningkatkan efisiensi operasional dan memperluas jangkauan pasar di era digital. Melalui sistem ini, proses pengelolaan data produk, stok, dan transaksi penjualan menjadi lebih terstruktur serta mudah diakses oleh pemilik maupun pelanggan. Penerapan sistem berbasis web memungkinkan pelanggan untuk melakukan pemesanan kapan saja dan di mana saja tanpa dibatasi ruang dan waktu, sehingga meningkatkan fleksibilitas serta kenyamanan dalam bertransaksi. Selain itu, sistem ini mendukung proses bisnis yang transparan dan efisien, karena setiap data penjualan tersimpan secara otomatis dan dapat dimonitor secara real-time. Dengan adanya sistem ini, SAponik *Farm* tidak hanya berhasil beradaptasi terhadap perkembangan teknologi informasi, tetapi juga memperkuat posisinya sebagai pelaku usaha pertanian modern yang responsif terhadap perubahan kebutuhan pasar dan perkembangan digital saat ini.

Lebih jauh, keberadaan aplikasi ini menjadi bentuk nyata dari transformasi digital yang tidak hanya meningkatkan produktivitas internal, tetapi juga memperluas peluang kolaborasi dengan pelanggan dan mitra bisnis. Meskipun sistem yang dibangun telah memberikan hasil yang signifikan, penelitian ini masih memiliki ruang untuk pengembangan lebih lanjut. Di masa mendatang, aplikasi *E-Commerce* SAponik *Farm* dapat diintegrasikan dengan sistem *payment gateway* untuk otomatisasi proses pembayaran, dikembangkan ke dalam bentuk aplikasi mobile agar lebih mudah diakses oleh pengguna, serta diperluas dengan fitur laporan keuangan dan analisis penjualan guna mendukung pengambilan keputusan yang lebih strategis. Dengan pengembangan berkelanjutan dan komitmen dalam penerapan teknologi, diharapkan sistem ini dapat menjadi model pengelolaan digital yang efisien, berkelanjutan, dan mampu meningkatkan daya saing usaha mikro di sektor pertanian berbasis hidroponik.

Referensi

- [1] D. Avriel, I. , B. Huda, S. S. Hilabi, and B. Priyatna, "Penerapan Desain UI/UX Pada Sistem Penjualan Berbasis Web Dengan Metode Desain Thinking," vol. 4, no. 2, 2024.
- [2] M. Jannah and A. A. Nugraha, "Pembuktian Tindak Pidana Bisnis Online terhadap Kasus Penipuan melalui Transaksi Elektronik (E-Commerce)," 2023. [Online]. Available: <https://ojs.staialfurqan.ac.id/jtm/>
- [3] M. I. Wiawan, R. Fadillah, and D. M. Yusup, "Sistem Informasi Pelayanan Administrasi Berbasis Website Dengan Metode Rup (Rational Unified Process) (Studi Kasus Kantor Urusan Agama Kecamatan Panjalu)," *Edusaintek: Jurnal Pendidikan, Sains dan Teknologi*, vol. 10, no. 3, pp. 758–776, May 2023, doi: 10.47668/edusaintek.v10i3.865.
- [4] N. A. Ahadiyah, "Perkembangan Teknologi Infomasi Terhadap Peningkatan Bisnis Online," *Interdisiplin: Journal of Qualitative and Quantitative Research*, vol. 1, no. 1, pp. 41–49, Dec. 2023, doi: 10.61166/interdisiplin.v1i1.5.
- [5] M. Rasid Ridho and Maydianto, "Rancang Bangun Sistem Informasi Point Of Sale Dengan Framework Codeigniter Pada Cv Powershop," 2021.
- [6] T. A. Hidayati, E. A. Widyantoro, and J. H. Ramadhani, "Perancangan Sistem Informasi Wirausaha Mahasiswa (Siwirma) Berbasis Web dengan Unified Modelling Language (UML)," *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 2, no. 4, pp. 86–107, Nov. 2023, doi: 10.55606/juprit.v2i4.2906.
- [7] S. E. T. Sami, S. Rahmawati, A. Prasetyo, and C. Cahyono, "Aplikasi Sistem Informasi Penjualan Pada Rumah Makan 'Jeng Tin' Menggunakan Database MySQL," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 4, no. 1, pp. 1–14, Jun. 2024, doi: 10.59395/janitra.v4i1.178.
- [8] M. Badrul, R. Ardy, S. Nusa Mandiri Jl Jatiwaringin Raya No, and K. Cipinang Melayu Jakarta Timur, "Penerapan Metode Waterfall pada Perancangan Sistem Informasi Pendaftaran Siswa Baru," 2021.
- [9] A. Patappari and N. Muhlisa, "Sistem Informasi Persediaan Barang Berbasis Web Pada Toko Throve Store Soppeng," *Jurnal Ilmiah Sistem Informasi dan Teknik Informatika (JISTI)*, vol. 6, no. 1, pp. 1–8, Apr. 2023, doi: 10.57093/jisti.v6i1.142.

- [10] A. Jolie, D. Dedrick, R. K. Sugeng, W. A. Lee, and A. Yulianto, “Aplikasi Sistem Manajemen Perpustakaan dengan Penerapan Pemrograman Berorientasi Objek,” *Telcomatics*, vol. 7, no. 2, Dec. 2022, doi: 10.37253/telcomatics.v7i2.7349.
- [11] B. Damanik, “Rancangan Sistem Informasi Smp Negeri 1 Tuhemberua Kabupaten Nias Utara Menggunakan Php Codeigniter,” *Jurnal Mahajana Informasi*, vol. 6, no. 1, 2023.
- [12] W. Apriyanti *et al.*, “Sosialisasi Penggunaan Internet Yang Sehat Bagi Anak-Anak Di Yayasan Domyadhu,” vol. 1, no. 1, 2022, [Online]. Available: <https://jurnal.portalpublikasi.id/index.php/AJP/index>
- [13] R. A. Aziz, A. Sansprayada, and K. Mariskhana, “Perancangan Sistem Administrasi Penjualan pada PT SurMoRin dengan Menggunakan PHP dan MYSQL,” *Jurnal Minfo Polgan*, vol. 13, no. 2, pp. 1641–1650, Oct. 2024, doi: 10.33395/jmp.v13i2.14148.
- [14] R. Wijayanti, Y. A. Indrastuti, U. Hudiana, D. O. Sumadya, R. F. Sarie, and M. Indrawati, “Sistem Informasi Penyewaan Lapangan Futsal Berbasis Web Dengan Metode Waterfall Menggunakan PHP,” *Jurnal ilmiah Sistem Informasi dan Ilmu Komputer*, vol. 5, no. 1, pp. 09–22, Mar. 2025, doi: 10.55606/juisik.v5i1.935.
- [15] Efendi, E., Wulandari, C., Siregar, I. A., Aulia, N., & Harahap, R. A. G. “Manajemen Database Sistem Organisasi Dakwah”. *Innovative: Journal Of Social Science Research*, 3(2), 1-10. 2023