

SISTEM INFORMASI PENJUALAN IKAN HIAS BERBASIS WEB DENGAN INTEGRASI PAYMENT GATEWAY DAN CHATBOT N8N PADA HOME SERVER STB HG680P

Mohammad Faisal, Rina Firliana, Aidina Ristyawan

Sistem Informasi, Universitas Nusantara PGRI Kediri

Jl. K.H. Achmad Dahlan No. 76, Kota Kediri, Jawa Timur, Indonesia

faisalariya009@gmail.com

ABSTRAK

UMKM ikan hias menghadapi kendala karena informasi produk dan stok, pencatatan pesanan, konfirmasi pembayaran, serta layanan pelanggan masih dikelola melalui kanal yang terpisah. Ketergantungan pada marketplace dan hosting komersial juga membatasi kendali terhadap branding, data pelanggan, dan biaya operasional. Penelitian ini bertujuan membangun sistem informasi penjualan ikan hias berbasis *web* yang mengintegrasikan *payment gateway* Midtrans dan *chatbot n8n* serta memanfaatkan STB HG680P sebagai *home server*. Pengembangan menggunakan model *Waterfall* melalui tahap analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Aplikasi dibangun dengan *Laravel*, *MySQL*, *Tailwind CSS*, dan *Docker* pada *CasaOS*, sedangkan akses publik disediakan melalui *Cloudflare Tunnel*. Hasil implementasi menunjukkan bahwa sistem dapat mengelola katalog, stok, keranjang, pesanan, pembayaran digital, perubahan status transaksi melalui *webhook*, dan layanan informasi pelanggan. Pengujian *Black Box* terhadap 12 skenario fungsional menunjukkan seluruh skenario berstatus valid, sehingga tingkat keberhasilan fungsionalitas sebesar 100%. Dengan batasan beban kerja ringan hingga menengah, STB HG680P dapat dimanfaatkan sebagai alternatif *hosting* mandiri berbiaya rendah bagi UMKM.

Kata kunci: Sistem Informasi, *Laravel*, *Payment gateway*, *Chatbot n8n*, *Home server*, STB HG680P

1. PENDAHULUAN

Digitalisasi penjualan membantu UMKM memusatkan informasi produk, stok, pesanan, transaksi, dan pelayanan pelanggan. Sistem penjualan berbasis *web* juga dapat memperluas media promosi serta menyediakan akses informasi tanpa dibatasi waktu dan lokasi [1], [2].

Pada UMKM ikan hias yang menjadi objek penelitian, permasalahan yang menjadi fokus pengembangan meliputi katalog dan ketersediaan stok yang belum dikelola dalam satu sistem, pencatatan pesanan yang tersebar pada beberapa kanal komunikasi, konfirmasi pembayaran yang memerlukan pemeriksaan manual, serta pertanyaan pelanggan mengenai jenis ikan, harga, stok, dan proses pemesanan yang berulang. Ketergantungan pada *marketplace* pihak ketiga juga membatasi pengelolaan branding dan data pelanggan, sedangkan *hosting* komersial menimbulkan biaya berulang. Kondisi tersebut berpotensi menyebabkan informasi stok tidak konsisten, pelayanan lebih lambat, dan riwayat transaksi sulit ditelusuri.

Untuk menjawab permasalahan tersebut, penelitian ini mengembangkan sistem informasi penjualan berbasis *web* menggunakan *Laravel* dan *MySQL*, mengintegrasikan *payment gateway* Midtrans untuk pembayaran digital, serta menerapkan *workflow automation n8n* sebagai *chatbot* layanan informasi. *Docker* digunakan untuk mengisolasi layanan aplikasi dan mempermudah *deployment*, sementara *Cloudflare Tunnel* menyediakan akses publik melalui koneksi keluar dari *server* lokal [3], [4], [5].

STB HG680P dipilih karena perangkat ini dapat dimanfaatkan kembali, berukuran ringkas, memiliki biaya awal rendah, dan mampu menjalankan sistem operasi *Linux* berbasis ARM serta layanan server ringan. *Raspberry Pi* memiliki ekosistem dan dukungan perangkat lunak yang lebih luas, tetapi memerlukan pembelian papan dan perlengkapan pendukung, *mini PC* menawarkan kinerja serta kompatibilitas x86 yang lebih tinggi, tetapi biaya pengadaan dan konsumsi dayanya cenderung lebih besar, sedangkan *VPS* memberikan infrastruktur publik yang dikelola penyedia, tetapi menimbulkan biaya berlangganan dan ketergantungan pada layanan eksternal. Dengan demikian, STB HG680P dipilih untuk skenario beban ringan hingga menengah dan *hosting* mandiri, bukan sebagai pengganti mutlak *mini PC* atau *VPS* pada kebutuhan trafik tinggi dan ketersediaan layanan yang kritis [6].

Kontribusi penelitian ini adalah integrasi penjualan, pembayaran, *chatbot*, dan *home server* berbiaya rendah dalam satu arsitektur yang ditujukan untuk kebutuhan digitalisasi UMKM ikan hias.

2. TINJAUAN PUSTAKA

2.1. Sistem Informasi Penjualan

Sistem informasi penjualan berbasis *web* mengintegrasikan data produk, persediaan, pelanggan, pesanan, dan transaksi agar proses operasional dapat dipantau secara lebih terstruktur. Penerapannya pada UMKM dapat mendukung promosi, mempercepat akses informasi, dan mengurangi ketergantungan pada pencatatan yang terpisah [1], [7].

2.2. *Laravel*

Laravel merupakan *framework PHP* yang menyediakan struktur pengembangan aplikasi *web*, routing, pengelolaan basis data, autentikasi, dan integrasi layanan pihak ketiga. Fitur tersebut digunakan untuk membangun modul katalog, keranjang, pesanan, dan administrasi sistem [8], [9].

2.3. *Payment gateway Midtrans*

Payment gateway menghubungkan aplikasi penjualan dengan kanal pembayaran digital. *Midtrans Snap* digunakan untuk menampilkan pilihan pembayaran, sedangkan *webhook* atau *payment notification* digunakan untuk mengirim perubahan status transaksi dari *Midtrans* ke aplikasi sehingga verifikasi pembayaran dapat diperbarui secara otomatis [10], [11].

2.4. *Chatbot dan Workflow Automation n8n*

n8n merupakan *platform workflow automation* yang menghubungkan *webhook*, logika proses, dan berbagai layanan komunikasi. Dalam penelitian ini, *n8n* digunakan untuk mengatur alur *chatbot* pada *web*, *Telegram*, dan *WhatsApp* sehingga pertanyaan umum pelanggan dapat direspons secara otomatis [12].

2.5. *Docker, CasaOS, dan Home server STB HG680P*

Docker menjalankan aplikasi dan dependensinya dalam *container* yang terisolasi. *CasaOS* digunakan sebagai antarmuka pengelolaan layanan pada *STB HG680P*. Studi terdahulu menunjukkan bahwa *STB HG680P* berbasis *Armbian* dapat menjalankan fungsi *home server* untuk beban ringan, meskipun demikian, keterbatasan *CPU*, *RAM*, penyimpanan, dan arsitektur *ARM* tetap perlu dipertimbangkan saat menempatkan beberapa layanan secara bersamaan [4], [6].

2.6. *Cloudflare Tunnel*

Cloudflare Tunnel menghubungkan *server* lokal dengan jaringan *Cloudflare* melalui koneksi keluar sehingga aplikasi dapat dipublikasikan tanpa membuka *port* masuk secara langsung. Pendekatan ini menyederhanakan akses *domain* publik, tetapi keamanan aplikasi, pengelolaan kredensial, pembaruan perangkat lunak, dan pencadangan data tetap menjadi tanggung jawab pengelola sistem [5].

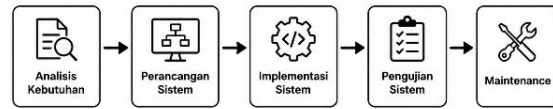
2.7. *Black Box Testing*

Black Box Testing mengevaluasi fungsi sistem berdasarkan masukan dan keluaran yang diharapkan tanpa menilai kode program secara internal. Metode ini digunakan untuk memeriksa fungsi autentikasi, pengelolaan produk, transaksi, *payment gateway*, *chatbot*, dan akses publik [13].

3. METODE PENELITIAN

Penelitian menggunakan *Software Development Life Cycle (SDLC)* dengan model *Waterfall* karena

ruang lingkup fungsi telah ditentukan sejak awal dan setiap tahapan dapat didokumentasikan secara berurutan. Tahapan yang digunakan meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan [14], [13]. Alur penelitian ditunjukkan pada Gambar 1.



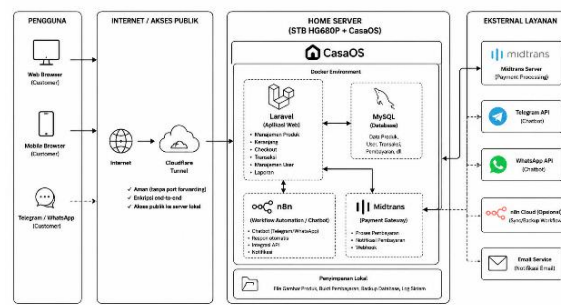
Gambar 1. Alur penelitian

3.1. Analisis Kebutuhan

Tahap analisis menetapkan kebutuhan pengguna dan administrator. Kebutuhan pengguna mencakup melihat katalog dan stok, mengelola keranjang, melakukan *checkout*, membayar melalui *Midtrans*, dan memperoleh informasi melalui *chatbot*. Kebutuhan administrator meliputi autentikasi, pengelolaan produk, stok, pesanan, transaksi, serta pemantauan status pembayaran. Kebutuhan nonfungsional mencakup akses responsif, pemisahan layanan melalui *container*, akses *domain* publik, dan pencadangan basis data.

3.2. Perancangan Sistem

Tahap perancangan menghasilkan rancangan antarmuka, struktur basis data, alur *checkout* dan notifikasi pembayaran, *workflow chatbot*, serta arsitektur deployment. *Laravel* berkomunikasi dengan *MySQL* untuk data aplikasi, *Midtrans* mengirimkan notifikasi pembayaran melalui *webhook*, dan *n8n* mengelola alur pesan *chatbot*. Arsitektur sistem ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur sistem

3.3. Implementasi Sistem

Implementasi dilakukan menggunakan *Laravel*, *MySQL*, *Tailwind CSS*, *Midtrans*, dan *n8n*. Pada tahap pengembangan, *Ngrok* digunakan sementara untuk memeriksa *webhook*. Setelah fungsi berjalan, aplikasi dan layanan pendukung ditempatkan dalam *Docker container* pada *STB HG680P* berbasis *Armbian* dan *CasaOS*. *Cloudflare Tunnel* kemudian digunakan untuk menghubungkan *domain* publik ke layanan *web* pada jaringan lokal.

3.4. Pengujian Sistem

Pengujian dilakukan menggunakan *Black Box Testing* terhadap 12 skenario yang mewakili fungsi *autentikasi*, pengelolaan produk, katalog, keranjang, *checkout*, *Midtrans* Snap, *webhook* pembayaran, pengelolaan transaksi, *chatbot*, dan akses publik. Persentase keberhasilan dihitung menggunakan rumus: $(\text{jumlah skenario valid} / \text{jumlah seluruh skenario}) \times 100\%$.

3.5. Pemeliharaan Sistem

Pemeliharaan mencakup pemantauan status *container* dan log aplikasi, pembaruan *dependency*, pencadangan basis data dan berkas produk, pemeriksaan kapasitas penyimpanan, serta pemulihan layanan ketika listrik atau jaringan terputus. Tahap ini diperlukan karena *home server* bergantung pada perangkat, koneksi internet, dan pengelolaan lokal.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Analisis Kebutuhan

Hasil analisis menetapkan dua kelompok pengguna, yaitu pelanggan dan administrator. Pelanggan memerlukan katalog, informasi stok, keranjang, *checkout*, pembayaran digital, dan bantuan informasi. Administrator memerlukan fungsi login, pengelolaan produk dan stok, pemantauan pesanan, serta perubahan status transaksi. Kebutuhan tersebut menjadi dasar modul yang diimplementasikan dan skenario pengujian pada tahap berikutnya.

4.2. Hasil Perancangan Sistem

Perancangan menghasilkan arsitektur terintegrasi sebagaimana Gambar 2. Browser pengguna mengakses aplikasi *Laravel* melalui *Cloudflare Tunnel*. *Laravel* mengelola data pada *MySQL*, mengirim permintaan transaksi ke *Midtrans*, menerima notifikasi pembayaran, dan berkomunikasi dengan *workflow n8n* untuk layanan *chatbot*. Pemisahan layanan dalam *container* memudahkan konfigurasi dan pengelolaan *dependency*.

4.3. Hasil Implementasi Sistem

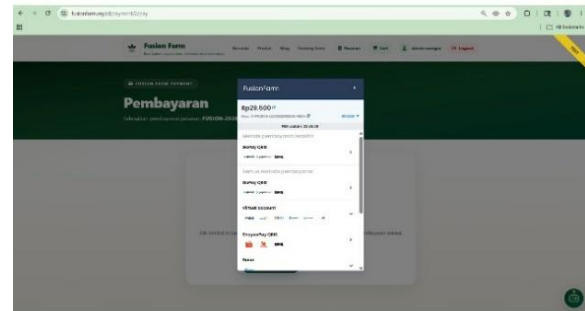


Gambar 3. Tampilan halaman utama website

Berdasarkan Gambar 3, halaman utama menampilkan informasi toko, kategori, katalog produk ikan hias, harga, dan navigasi menu. Informasi yang terpisah mengurangi kebutuhan pelanggan untuk

menanyakan katalog dan stok melalui kanal terpisah. Antarmuka dibangun dengan *Tailwind CSS* agar dapat menyesuaikan ukuran layar perangkat.

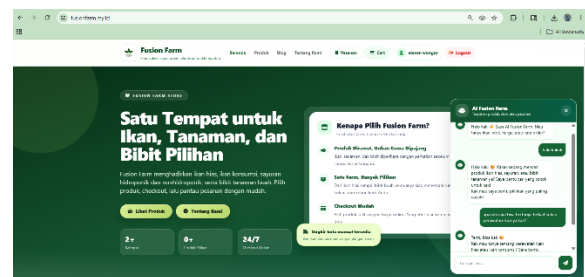
Modul *checkout* diintegrasikan dengan *Midtrans* Snap. Setelah pelanggan mengonfirmasi pesanan, aplikasi membentuk transaksi dan menampilkan pilihan metode pembayaran. Implementasi halaman pembayaran ditunjukkan pada Gambar 4.



Gambar 4. Implementasi *payment gateway* *Midtrans*

Berdasarkan Gambar 4, *Midtrans* Snap dapat ditampilkan setelah proses *checkout*. Notifikasi pembayaran yang diterima melalui *webhook* digunakan untuk menyelaraskan status transaksi pada aplikasi. Mekanisme ini mengurangi pemeriksaan pembayaran secara manual; namun endpoint *webhook* tetap harus dilindungi dengan verifikasi signature key dan pencatatan log [11].

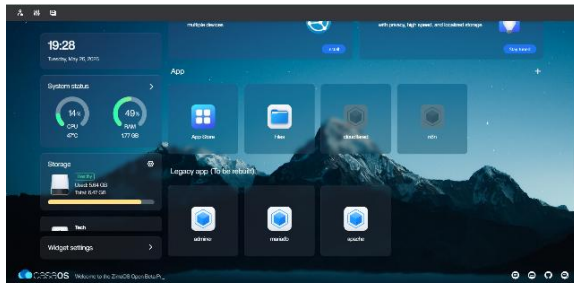
Layanan pelanggan diimplementasikan melalui *workflow n8n* yang menerima pesan dari *webhook* *web*, Telegram, atau WhatsApp, memproses aturan atau sumber informasi, kemudian mengirimkan respons kepada pengguna. Implementasi *chatbot* pada website ditunjukkan pada Gambar 5.



Gambar 5. Implementasi *chatbot* *n8n* pada web

Berdasarkan Gambar 5, *chatbot* dapat menerima pertanyaan dan mengirim respons informasi secara otomatis. Integrasi ini mendukung penanganan pertanyaan berulang, tetapi ketepatan jawaban tetap bergantung pada rancangan *workflow*, data sumber, dan pengelolaan kondisi ketika pertanyaan tidak dikenali [15].

Aplikasi kemudian di-deploy pada *STB HG680P* berbasis *CasaOS*. *Docker* menjalankan layanan *Laravel*, *MySQL*, *n8n*, dan *Cloudflare Tunnel* secara terpisah. Implementasi *home server* ditunjukkan pada Gambar 6.

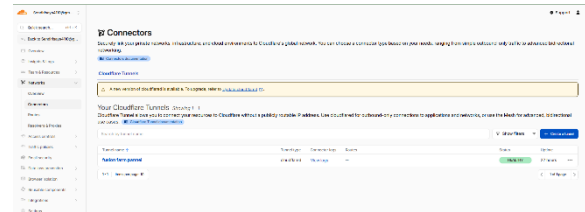


Gambar 6. Implementasi *home server* STB HG680P menggunakan CasaOS

Berdasarkan Gambar 6, layanan dapat dijalankan dan dikelola melalui CasaOS. Hasil tersebut menunjukkan kelayakan fungsional STB untuk beban aplikasi penelitian. Pemilihan STB memberikan keuntungan biaya awal dan pemanfaatan ulang perangkat, tetapi sumber daya perangkat lebih terbatas dibandingkan *mini PC* dan *VPS*. Oleh sebab itu, skalabilitas perlu dievaluasi melalui pengujian beban, penggunaan *CPU/RAM*, suhu, dan uptime pada penelitian selanjutnya [6].

Cloudflare Tunnel digunakan pada tahap *deployment* untuk menyediakan akses melalui *domain* publik tanpa *port forwarding* langsung. Hasil

implementasi akses publik ditunjukkan pada Gambar 7.



Gambar 7. Implementasi Cloudflare Tunnel pada website

Berdasarkan Gambar 7, website dapat diakses melalui *domain* yang dipetakan ke Cloudflare Tunnel. Tunnel mengurangi kebutuhan membuka *port* masuk pada *router*, tetapi tidak menggantikan kontrol keamanan pada aplikasi, *autentikasi* administrator, pengelolaan rahasia, pembaruan sistem, dan pencadangan data [13].

4.4. Hasil Pengujian Sistem

Pengujian *Black Box* dilakukan untuk membandingkan keluaran aktual dengan hasil yang diharapkan pada fungsi utama. Rekapitulasi pengujian ditunjukkan pada Tabel 1.

Tabel 1. Hasil *Black Box Testing*

No	Skenario	Hasil yang Diharapkan	Status
1	Login admin valid	Dashboard ditampilkan	Valid
2	Login admin tidak valid	Akses ditolak dan pesan tampil	Valid
3	Kelola produk	Data dapat ditambah, diubah, dan dihapus	Valid
4	Tampil katalog dan stok	Data sesuai basis data	Valid
5	Kelola keranjang	Jumlah item dapat diperbarui/dihapus	Valid
6	Checkout	Pesanan dan total transaksi terbentuk	Valid
7	Midtrans Snap	Pilihan pembayaran ditampilkan	Valid
8	Webhook Midtrans	Status pembayaran diperbarui	Valid
9	Kelola transaksi admin	Pesanan dapat dipantau/diperbarui	Valid
10	Chatbot web	Pertanyaan menerima respons	Valid
11	Webhook Telegram/WhatsApp	Pesan diterima dan respons dikirim	Valid
12	Cloudflare Tunnel	Website dapat diakses melalui <i>domain</i>	Valid

Seluruh 12 skenario memperoleh status valid. Tingkat keberhasilan fungsionalitas = $(12/12) \times 100\% = 100\%$. Hasil ini menunjukkan bahwa fungsi utama telah menghasilkan keluaran sesuai kebutuhan yang diuji. Pengujian *Black Box* membuktikan kesesuaian fungsi, tetapi belum mengukur performa *server*, kemampuan menangani pengguna serentak, keamanan aplikasi, ketahanan ketika listrik atau internet terputus, maupun kualitas jawaban *chatbot*. Oleh karena itu, hasil 100% tidak dapat diartikan sebagai jaminan bahwa sistem bebas dari kegagalan pada seluruh kondisi operasional.

4.5. Hasil Pemeliharaan dan Evaluasi

Pemeliharaan dilakukan melalui pemantauan *container* dan log, pencadangan basis data, pemeriksaan kapasitas penyimpanan, serta pembaruan komponen aplikasi. Evaluasi menunjukkan bahwa ketergantungan pada listrik, jaringan internet, dan

sumber daya terbatas STB merupakan risiko operasional. Mitigasi yang disarankan meliputi backup berkala ke media berbeda, penggunaan *UPS*, pembatasan akses administrator, rotasi kredensial, dan pengujian pemulihan data.

5. KESIMPULAN DAN SARAN

Sistem informasi penjualan ikan hias berbasis *web* berhasil mengintegrasikan pengelolaan produk dan stok, keranjang, *checkout*, transaksi, pembayaran digital *Midtrans*, *chatbot* n8n, serta akses publik melalui Cloudflare Tunnel. Aplikasi dan layanan pendukung dapat dijalankan dalam *Docker container* pada STB HG680P berbasis CasaOS. Hasil *Black Box Testing* menunjukkan 12 dari 12 skenario berstatus valid dengan tingkat keberhasilan fungsionalitas 100%, mencakup autentikasi, pengelolaan produk, katalog, keranjang, *checkout*, *Midtrans Snap*, *webhook* pembayaran, transaksi admin, *chatbot*, dan akses

domain publik. Hasil tersebut menunjukkan kesesuaian fungsi terhadap kebutuhan yang diuji, tetapi belum mencakup pengujian beban, uptime, keamanan, konsumsi sumber daya, atau akurasi jawaban *chatbot*. STB HG680P sesuai digunakan sebagai alternatif *home server* untuk beban ringan hingga menengah; kebutuhan trafik tinggi dan ketersediaan kritis tetap memerlukan infrastruktur yang lebih kuat. Penelitian selanjutnya disarankan menambahkan pengujian performa, keamanan, *failover*, pencadangan otomatis, dan evaluasi pengguna.

DAFTAR PUSTAKA

- [1] Robby Novianto And Maryam, "Sistem Informasi Penjualan Berbasis Web Pada UMKM R-DUA Lencana Kudus," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, Vol. 10, No. 2, Pp. 35–42, Nov. 2022, Doi: 10.21063/Jtif.2022.V10.2.35-42.
- [2] R. Firliana Et Al., "Perancangan Sistem Stok Barang Berbasis Website Pada Konter Handphone," *Jurnal Nusantara Of Engineering*, Vol. 5, No. 2, 2022, Doi: <https://doi.org/10.29407/Noe.V5i2.18867>.
- [3] "Laravel LLC, 'Laravel 12.X Documentation,'" <https://laravel.com/docs/12.x>.
- [4] Docker Inc., "Docker Documentation: Containers And Multi-Container Applications," <https://docs.docker.com>.
- [5] Cloudflare Inc., "Cloudflare Tunnel," <https://developers.cloudflare.com/cloudflare-one/networks/connectors/cloudflare-tunnel/>.
- [6] M. Gilang Ramandani And A. Zafia, "Pemanfaatan STB Android Sebagai Home Server Untuk Jaringan Rumah," 2025.
- [7] P. Sokibi, M. Irfan, And C. Nas, "Rancang Bangun Katalog Elektronik Kerajinan Rotan Berbasis Website Pada Cv Aesthetic Rattan Menggunakan Metode User Centered Design," 2024. Doi: 10.36040/Jati.V8i6.11341.
- [8] C. Gibran, A. R. Dewi, And E. Hadinata, "Implementasi Framework Laravel Untuk Pengembangan Website Penjualan Ayam Potong Dengan Pemanfaatan Midtrans Menggunakan Metode Fast," *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, Vol. 7, No. 1, Pp. 246–253, 2024, Doi: <https://doi.org/10.55338/Jikomsi.V7i1.2920>.
- [9] B. Hakim, "Sistem Informasi Penjualan Berbasis Web Dengan Framework Laravel," 2026. Doi: <https://doi.org/10.36040/Jati.V10i1.16742>.
- [10] M. Al Biruni, A. Faisol, And N. Vendyansyah, "Penerapan Rest Api Dan Integrasi Midtrans Sebagai Payment Gateway Pada Platform Pelatihan Online (Studi Kasus: Bisabacakitab)," 2023. Doi: 10.36040/Jati.V7i4.7554.
- [11] A. Sudianto, B. A. C. Permana, Muhammad Wasil, And Harianto, "Penerapan Sistem Payment Gateway Pada E-Commerce Sebagai Upaya Peningkatan Penjualan," *Infotek: Jurnal Informatika Dan Teknologi*, Vol. 8, No. 1, Pp. 271–279, Jan. 2025, Doi: 10.29408/Jit.V8i1.28323.
- [12] Rahma Et Al. N.D., "Implementasi Chatbot AI Berbasis N8n Untuk Layanan Informasi Administrasi Akademik Departemen Teknik Elektro Dan Informatika," 2025, Doi: <https://doi.org/10.26418/Jp.V11i3.101635>.
- [13] F. Nurdiansyah, E. Daniati, And A. Ristyawan, "Pengembangan Sistem Informasi Kasir Apotek Dengan Metode Waterfall," *EDUSAINTEK: Jurnal Pendidikan, Sains Dan Teknologi*, Vol. 9, No. 3, Pp. 752–773, Aug. 2022, Doi: 10.47668/Edusaintek.V9i3.550.
- [14] J. Alif Ramadhan, D. Tresya Haniva, And A. Suharso, "Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, Dan Hybrid," 2023. Doi: 10.26740/Jieet.V7n1.P36-42.
- [15] Aditia Ramadhani, Maulana Dwi Yantoro, Muhammad Farhan Akmal, Muhammad Mahfud, And Fauzi, "Chatbot Otomatis Dengan N8n Dan Ai Untuk Analisis Data Dan Pelaporan Hasil," *Jurnal Riset Teknik Komputer*, Vol. 2, No. 2, Pp. 18–23, Jun. 2025, Doi: 10.69714/X1p94182