IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya

OLEH

MUHAMMAD ISNAINI
C2155201110

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2025

IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya

Oleh:

MUHAMMAD ISNAINI
C2155201110

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER(STMIK) PALANGKARAYA
2025

PERSETUJUAN

IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*

Tugas Akhir Ini Telah Disetujui Dan Disahkan

Pembimbing I,	Pembimbing II,
Ir. Abdul Hadi, S.T., M.Kom.	Rudini,M.Pd.
NIK.198505102021101	NIK.198709172015105

Mengetahui Ketua STMIK Palangkaraya,	Mengesahkan Ketua Prodi Teknik Informatika,
Ir. Hj. Siti Maryamah, M.M	Ir. Abdul Hadi, S.T., M.Kom
NIK. 196902021995004	NIK. 198505102021101

PENGESAHAN

IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*

Tugas Akhir ini telah Diujikan, Dinilai, dan Disahkan
Oleh Tim Penguji pada Tanggal 18 November 2025

Tim Penguji Tugas Akhir :

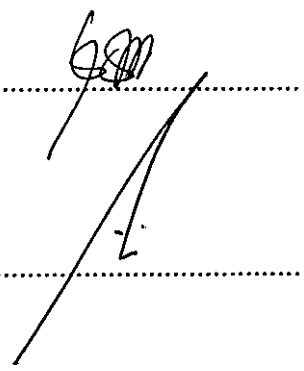
1. Amaya Andri Damaini, S.Kom., M.T. ..... 
   Ketua

2. Elok Faiqotul Himmah, S.Si., M.Sc ..... 
   Sekretaris

3. Elia Zakharia, M.T ..... 
   Anggota

4. Ir. Abdul Hadi, S.T., M.Kom .....
   Anggota

5. Rudini, M.Pd .....
   Anggota



PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama Mahasiswa	:	Muhammad Isnaini
N I M	:	C2155201110

menyatakan bahwa Tugas Akhir dengan judul :

IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN WATERMARKING DAN *ENKRIPSI AES*

Adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 17 Desember 2025

Membuat Pernyataan,



MUHAMMAD ISNAINI

MOTTO DAN PERSEMBAHAN

Seperti bunga dandelion yang hancur dan beterbangan, ia akan selalu menemukan harapan untuk tumbuh kembali, dimanpun angin membawanya.

Sebagai Ungkapan Terimakasih, Tugas akhir ini Penulis Persembahkan kepada:

- Ayah dan Ibu, pahlawan tanpa senjata, yang berjuang tanpa pamrih, dan ratu tanpa mahkota, yang cintanya tak pernah pudar. Terima kasih telah menjadi alasan dibalik setiap langkah dan semangatku hingga titik ini.

- Kakakku, malaikat tanpa sayap yang selalu menjaga, membimbing, dan menguatkanku tanpa pernah lelah. Kalian adalah anugrah yang tak ternilai harganya.

- Seluruh dosen STMIK palangkaraya yang telah membimbing, menginspirasi, dan telah membuka wawasan selama perjalanan studi ini. Terima kasih atas ilmu, kesabaran, dan dedikasi Bapak/Ibu dalam membentuk kami menjadi insan yang lebih siap menghadpi masa depan.

- Kepada dosen pembimbing Bapak Abdul Hadi, ST., M.Kom., dan Bapak Rudini, M.Pd., atas Bimbingan, motivasi, arahan yang tak ternilai dalam proses penyusunan tugas akhir ini. Terima kasih telah meluangkan waktu untuk membimbing saya hingga akhir.

- Serta untuk teman teman seperjuangan Angkatan 2021, yang telah menjadi bagian cerita indah selama masa kuliah.

INTISARI

Muhammad Isnaini, C2155201110, 2025. *Implementasi Keamanan Penjualan Baju Adat Berbasis Web Menggunakan Watermarking dan Enkripsi AES,* Pembimbing I Abdul Hadi, ST., M.Kom., Pembimbing II Rudini, M.Pd.

Toko baju adat Bawi Lamus Palangkaraya memanfaatkan platform digital untuk memperluas pasar, namun menghadapi tantangan serius terkait keamanan aset digital, khususnya gambar produk yang rentan dicuri atau dipalsukan dan data pelanggan yang berisiko bocor. Untuk mengatasi hal ini, penelitian ini mengimplementasikan sistem keamanan terpadu: menggunakan teknologi *Watermarking* untuk melindungi keaslian gambar produk dan enkripsi *Advanced Encryption Standard* (AES) untuk mengamankan data pelanggan, yang juga merupakan bagian dari upaya menjaga nilai budaya lokal dari eksploitasi digital.

Pengembangan sistem dilakukan dengan menggunakan metode *Research and Development* (R&D) yang mencakup tahapan potensi dan masalah, pengumpulan data, desain produk, validasi desain, revisi desain, uji coba produk, revisi produk, hingga produk akhir. Data diperoleh melalui observasi dan wawancara dengan admin Toko Baju Adat Bawi Lamus Palangkaraya, kemudian dilakukan perancangan sistem dengan menggunakan pendekatan terstruktur serta implementasi menggunakan bahasa pemograman codeigniter dan PHP. Validasi sitem dilakukan oleh ahli dan pengguna melalui uji coba langsung untuk mengetahui tingkat kemananya.

Hasil Pengujian *Blackbox* menunjukan bahwa sistem berhasil memberikan tanda kepemilikan hak cipta produk dan keamanan data pembeli dalam bertransaksi. Dengan implementas ini, terbukti bahwa hak cipta produk dan keamanan data dalam bertransaksi di toko baju adat dapat ditingkatkan serta lebih efektif dan terkontrol.

Kata kunci *Watermarking* dan AES, Keamanan Aset Digital, Baju Adat Bawi Lamus, *Research and Development* (R&D), Perlindungan Hak Cipta.

ABSTRACT

Muhammad Isnaini, **C2155201110**, **2025**. *Implementation of Web-Based Traditional Clothing Sales Security Using Watermarking and AES Encryption*, *Supervisor I* Abdul Hadi, ST., M.Kom., *Supervisor II* Rudini, M.Pd.

The Bawi Lamus Traditional Clothing Store in Palangkaraya utilizes digital platforms to expand its market but faces serious challenges concerning digital asset security, particularly product images prone to theft or forgery and customer data at risk of leakage. To address this, this study implements an integrated security sistem: using Watermarking technology to protect the authenticity of product images and Advanced Encryption Standard (AES) encryption to secure customer data, which is also part of an effort to safeguard local cultural values from digital exploitation.

The sistem development was conducted using the Research and Development (R&D) method, covering the stages of potential and problems, data collection, product design, design validation, design revision, product trial, product revision, up to the final product. Data was obtained through observation and interviews with the administrator of the Bawi Lamus Traditional Clothing Store in Palangkaraya. The sistem was designed using a structured approach and implemented using the Codeigniter framework and PHP programming language. Sistem validation was carried out by experts and users through direct testing to determine its security level.

The Blackbox Testing results show that the sistem successfully provides a copyright ownership mark for products and security for customer transaction data. With this implementation, it is proven that product copyright and transaction data security in the traditional clothing store can be enhanced, becoming more effective and controlled.

Keywords: Watermarking and AES, Digital Asset Security, Baju Adat Bawi Lamus, Research and Development (R&D), Copyright Protection.

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai sivitas akademik Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya, saya yang bertanda tangan dibawah ini:

Nama : Muhammad Isnaini
NIM : C2155201110
Program Studi : Teknik Informatika
Jenis Karya : Laporan Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya, Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*

Dengan Hak Bebas Royalti Non Eksklusif ini Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya penulis/pencipta dan sebagai pemilik Hak Cipta.
Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Palangka Raya
Pada tanggal : 17 Desember 2025

Yang menyatakan

Muhammad Isnaini

KATA PENGANTAR

Alhamdulillah penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena dengan karunia-nya sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul " IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS *WEB* MENGGUNAKAN *WATERMARKING* DAN *ENKRIPSI AES*". Tugas Akhir ini disusun untuk memenuhi syarat dalam memeperoleh Kompoter di Sekolah Tinggi Menajemen Informatika dan Kompoter (STMIK) Palangkaraya.

Pada kesempatan ini penulis juga menyampaikan terimakasih yang sebesar besarnya kepada semua pihak yang telah memberikan bimbingan, motivasi, semangat, dalam menyelasaikan penulisan Tugas Akhir ini, yaitu kepada :

1. Bapak Stevano Adjie selaku pemilik toko baju Adat Bawi Lamus Palangkaraya,Telah menerima dan memberikan izin penelitian sebagai tempat penelitian.
2. Ibu Rika selaku admin yang bersedia menjadi narasumber pada penelitian ini.
3. Abdul Hadi, S.T., M.Kom. selalku dosen pembimbing I yang meluangkan waktunya untuk memberikan bimbingan dalam penyelesain Tugas Akhir ini.
4. Rudini M.Pd. selaku dosen pembimbing II dalam penulisan Tugas Akhir ini, yang juga banyak memberian saran dan juga masukan dalam Penulisan Tugas Akhir ini. Penulis mengharapkan masukan, saran dan kritik yang bersifat membangun untuk penyempurnaan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca terutam untuk menambah ilmu pengetahuan.

Palangka Raya, 17 Desember 2025

Penulis

Muhammad Isnaini

DAFTAR ISI

PERSETUJUAN ..... **iii**
PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN ..... **v**
MOTTO DAN PERSEMBAHAN ..... **vi**
INTISARI ..... **vii**
ABSTRAK ..... **viii**
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR ..... **ix**
KATA PENGANTAR ..... **x**
DAFTAR ISI ..... **xi**
DAFTAR TABEL ..... **xiv**
DAFTAR GAMBAR ..... **xiv**
DAFTAR LAMPIRAN ..... **xv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Dan Manfaat Penelitian 3
1.5 Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA **5**
2.1 Kajian Teori 5
2.1.1 *Implementasi* 5
2.1.2 *Watermarking* 5
2.1.3 *E-commerce* 6
2.1.4 Baju Adat 7
2.1.5 *Web* 7
2.1.6 *AES (Advanced Encryption Standard)* 8
2.1.7 Pemodelan Berorientasi Objek Menggunakan *Unified Modeling Language (UML)* 9
2.1.8 *MYSQL* 11
2.1.9 *Codeigniter* 12
2.1.10 *Draw.io* 12
2.1.11 *Mode Operasi AES CBC (Chiper Block Chaining)* 13
2.1.12 *R&D (Research And Development)* 13
2.2 Penelitian Yang Relevan 14
BAB III METODOLOGI PENELITIAN **18**
3.1 Tinjuan Umum 18
3.2 Jenis Penelitian 19
3.3 Desain Penelitian 19
3.4 Teknik Pengumpulan Data 20
3.5 Analisis Kebutuhan 21
3.5.1 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak Untuk Menjalankan Kebutuhan Sistem 22
3.6 Desain Sistem 22
3.6.1 Desain Proses 22
3.6.2 Desain Enkripsi AES 33

3.6. Desain Interface 35
BAB IV HASIL DAN PEMBAHASAN 41
4.1 Hasil 41
4.1.1 Gambaran Umum Sistem 41
4.1.2 Proses *Enkripsi AES 128* 43
4.1.3 Proses *Dekripsi AES 128* 44
4.1.4 Proses *Watermark* 47
4.1.5 Penggunaan *AES* di Dalam *Controller* 50
4.1.6 Penggunaan *Watermark* di Dalam *Controller* 52
4.1.7 Fitur dan Halaman *Admin* 64
4.1.8 Fitur dan Halaman Pengguna 72
4.1.9 Hasil Uji Coba 78
4.1.10 Pengujian Sistem 84
4.2 Pembahasan 84
4.2.1 Evaluasi Sistem 85
4.2.3 Kelebihan dan Kekurangan Sistem 85
BAB V PENDAHULUAN 87
5.1 Kesimpulan 87
5.2 Saran 88
DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2. 1 Tabel Putaran Algoritma AES tiap blok 8
Tabel 2. 2 *Use Case Diagram* 10
Tabel 2. 3 *Activity Diagram* 11
Tabel 2. 4 Penelitian yang Relevan 16
Tabel 3. 1 Kebutuhan Perangkat Lunak 22
Tabel 3. 2 Kebutuhan Perangkat Keras 22
Tabel 3. 3 Kebutuhan Perangkat Keras 36
Tabel 4. 1 Hasil Cropping 77
Tabel 4. 2 Pengujian Uji Coba Black Box 78

DAFTAR GAMBAR

Gambar 2. 1 Proses Enkripsi Algoritma AES-128 9
Gambar 3. 1 *Use Case Diagram Admin* 23
Gambar 3. 2 *Use Case Diagram Umum* 24
Gambar 3. 3 *Activity Diagram Sistem* 26
Gambar 3. 4 *Activity Diagram Enkripsi* 27
Gambar 3. 5 *Activity Diagram Dekripsi* *29*
Gambar 3. 6 *Activity Diagram Watermarking* 30
Gambar 3. 7 *Flowchart Enkripsi* Data Transaksi 31
Gambar 3. 8 *Flowchart Dekripsi* Data Transaksi 32
Gambar 3. 9 Desain Halaman Login 36
Gambar 3. 10 Desain Halaman *Dashboard* 36
Gambar 3. 11 Desain Proses *Watermarking* 37
Gambar 3. 12 Desain Data Transaksi 38
Gambar 3. 13 Desain Secret Key 39
Gambar 3. 14 Desain *Enkripsi* Database 40
Gambar 3. 15 Desain Tambah Produk 40
Gambar 4. 1 Login Admin 53
Gambar 4. 2 *Dashboard* 54
Gambar 4. 3 Transaksi 55
Gambar 4. 4 Detail Transaksi 56
Gambar 4. 5 Data Produk 57
Gambar 4. 6 Form Tamba Produk 58
Gambar 4. 7 Form Data Produk 59
Gambar 4. 8 Form Edit Produk 60
Gambar 4. 9 Data Kategori 60
Gambar 4. 10 Form Detail Kategori 61
Gambar 4. 11 Form Edit Kategori 62
Gambar 4. 12 Konfigurasi Pengaturan Akun 63
Gambar 4. 13 Konfigurasi *Watermark* 64
Gambar 4. 14 Halaman Utama Pelanggan 64
Gambar 4. 15 Halaman Produk Terbaru 65
Gambar 4. 16 Halaman Produk 66
Gambar 4. 17 Halaman Detail Produk 66
Gambar 4. 18 Halaman Login 67
Gambar 4. 19 Halaman Registrasi 68
Gambar 4. 20 Halaman Keranjang 69
Gambar 4. 21 Halaman Pesanan 69
Gambar 4. 22 Halaman Pembayaran 70
Gambar 4. 23 Halaman Riwayat Belanja 71
Gambar 4. 24 Daftar Pesanan 71
Gambar 4. 25 Detail Pesanan 72
Gambar 4. 26 Uji Enkripsi Nomor Telepon 72
Gambar 4. 27 Uji Enkripsi Alamat Chekout 72
Gambar 4. 28 Uji Dekripsi 73
Gambar 4. 26 Percobaan Croping 77

DAFTAR LAMPIRAN

Lampiran 1 Proses enkripsi AES dan Watermarking
Lampiran 2 Surat Tugas
Lampiran 3 Kartu kegiatan konsultasi tugas akhir
Lampiran 4 Surat tugas seminar proposal
Lampiran 5 Surat tugas sidang tugas akhir
Lampiran 6 Kartu kegiatan konsoltasi seminar proposal tugas akhir
Lampiran 7 Surat izin penelitian
Lampiran 8 Hasil uji coba black box
Lampiran 9 Dokomentasi wawancara

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi telah mendorong pelaku usaha, termasuk penjual baju adat, untuk memanfaatkan *platform digital* dalam memasarkan produknya. Penjualan baju adat secara daring memungkinkan pelaku *UMKM* (Usaha Mikro, Kecil, Menengah) menjangkau pasar yang lebih luas, baik nasional maupun internasional. Namun, di balik peluang tersebut, terdapat tantangan serius dalam hal keamanan, terutama pada perlindungan gambar produk dan data pelanggan yang rawan disalahgunakan di dunia *digital*.

Gambar produk baju adat yang ditampilkan di *website* sangat rentan terhadap pencurian, penyalahgunaan, dan pemalsuan. Di sisi lain, data informasi pelanggan yang tidak terlindungi dapat berisiko bocor dan menimbulkan kerugian. Oleh karena itu, diperlukan sistem keamanan yang mampu melindungi gambar serta data penting secara menyeluruh. Teknologi *Watermarking* dapat digunakan untuk menjaga keaslian gambar produk, sementara *Enkripsi AES* (*Advanced Encryption Standard*) dapat mengamankan data pelanggan.

Penelitian ini dilakukan di toko baju adat Bawi Lamus Palangkaraya yang berlokasi di Jalan Beliang Induk, yang telah menjalankan penjualan secara *online* dan menghadapi langsung tantangan tersebut. Toko ini dipilih karena aktif dalam memanfaatkan teknologi serta berkomitmen terhadap pelestarian budaya melalui produk yang dijual. Implementasi sistem keamanan berbasis *Watermarking* dan *Enkripsi AES* di toko ini tidak hanya bertujuan melindungi data dan gambar produk,

tetapi juga menjadi bagian dari upaya menjaga nilai budaya lokal dari eksploitasi yang tidak bertanggung jawab di era *digital*.

1.2 Perumusan Masalah

Bagaimana mengimplementasikan teknik *Watermarking* dan *Enkripsi AES* untuk meningkatkan keamanan penjualan baju adat di Toko Bawi Lamus Palangka Raya?.

1.3 Batasan Masalah

Agar lebih terfokus dan mencapai tujuan yang diinginkan, pembahasan dibatasi pada hal-hal berikut :

a. Penelitian ini hanya membahas implementasi *Watermarking* dengan jenis *watermark visible* sebagai metode untuk melindungi keaslian gambar produk baju adat pada sistem penjualan berbasis *web*.
b. Penelitian ini fokus pada penggunaan *Enkripsi AES 128* untuk mengamankan data transaksi dan informasi pelanggan dalam sistem penjualan *online*.
c. Penerapan sistem keamanan ini khusus dilakukan pada toko baju adat Bawi Lamus Palangkaraya saja.
d. Aplikasi penjualan baju adat yang dibangun berbasis *web* (atau *mobile*, sesuaikan dengan kebutuhan) dan menggunakan *framework* tertentu yaitu *CodeIgniter* dan *HTML*.

1.4 Tujuan Dan Manfaat Penelitian

a. Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan teknik *watermarking* dan *enkripsi AES* guna meningkatkan keamanan data pada sistem penjualan baju adat di Toko Bawi Lamus Palangka Raya.

b. Manfaat Penelitian

1) Manfaat bagi Bawi Lamus Palangkaraya

Penelitian ini mempermudah transaksi jual beli dan meperkuat pengamanan data dari penjual dan pembeli.

2) Manfaat bagi Penulis

Penelitian ini memberikan manfaat dalam mengasah kemampuan analisis dan penerapan teknologi keamanan informasi, khususnya dalam pengembangan sistem berbasis *web*. Melalui studi ini, penulis dapat memperdalam pemahaman tentang penerapan teknologi *Watermarking* dan *Enkripsi AES*, serta bagaimana kedua metode tersebut diintegrasikan dalam konteks nyata untuk menyelesaikan permasalahan keamanan *digital*.

3) Manfaat bagi STMIK Palangkaraya

Hasil Penelitian ini diharapkan dapat membantu penulis menyusun materi yang relevan untuk referensi yang ada di perpustakaan STMIK Palangkaraya. Selain itu, implementasi ini juga akan menjadi sumber pelajaran yang akan dapat digunakan oleh penulis lain sebagai referensi atau perbandingan literatur.

1.5 Sistematika Penulisan

Adapun sistematika penulisan dalam penulisan tugas akhir ini terdiri dari beberapa bab dan masing-masing bab membahas dan menguraikan pokok permasalahan yang berbeda, sebagai Gambaran disini penulis menyertakan garis-garis besarnya yaitu :

BAB I PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan tentang tinjauan pustaka yang di ambil dari penelitian yang relevan beserta susunan kajian teori yang disesuaikan dengan tema Tugas Akhir.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang tahapan yang akan dilakukan peneliti dalam mengumpulkan informasi atau data yang dibutuhkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan hasil dari penelitian dan pembahasan dari hasil yang didapat

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Teori

2.1.1 Implementasi

Implementasi mengacu pada tindakan untuk mencapai tujuan-tujuan yang telah ditetapkan dalam suatu keputusan. Tindakan ini berusaha untuk mengubah keputusan-keputusan tersebut menjadi pola-pola operasional serta berusaha mencapai perubahan-perubahan besar atau kecil sebagaimana yang telah diputuskan sebelumnya. *Implementasi* dipandang sebagai rangkaian tindakan terencana yang menerapkan konsep atau kebijakan ke praktik lapangan, meliputi persiapan, pelaksanaan, dan evaluasi agar tujuan tercapai (Sari,2021).

2.1.2 Watermaking

Watermarking adalah teknik penyisipan informasi tersembunyi ke dalam suatu media digital seperti gambar, video, atau dokumen, dengan tujuan melindungi hak cipta, menandai kepemilikan, atau menjaga integritas data. Informasi yang disisipkan disebut *watermark*, bisa berupa teks, logo, atau kode *digital* tertentu. Berbeda dengan enkripsi yang membuat data tidak dapat dibaca tanpa kunci, *Watermarking* justru menempelkan identitas pada media sehingga media tetap bisa dilihat atau digunakan seperti biasa, namun memiliki tanda khusus yang sulit dihilangkan tanpa merusak file aslinya.

Visible Watermarking adalah salah satu jenis *watermark* yang tampil secara jelas dan terlihat langsung pada media, misalnya logo transparan di pojok gambar

atau tulisan "*CONFIDENTIAL*" di tengah dokumen. Tujuan utama *visible Watermarking* adalah memberikan perlindungan visual agar orang yang melihat media tersebut mengetahui pemilik atau sumber asli file. Selain itu, *watermark* yang terlihat membuat orang enggan menggunakan atau menggandakan file secara ilegal, karena *watermark* menandai identitas dengan cara yang tidak bisa disangkal. Meskipun terlihat, desainnya biasanya dibuat semi-transparan agar tidak merusak kualitas *visual* keseluruhan, namun tetap cukup kuat untuk menunjukkan kepemilikan. *Visible Watermarking* dipilih pada kasus di mana pencegahan penyalahgunaan lebih penting daripada penyembunyian informasi.

Watermark merupakan gambaran besar alur bagaimana pengguna digital mengelola data dengan tujuan memberi tanda agar data tersebut tidak mudah diplagiasi maupun disalin kemudian disebarluaskan melalui pasar digital yang ada oleh pihak yang tidak berwenang (Yuana, Prihantu,2022).

2.1.3 E-commerce

E-commerce (*Electronic Commerce*) adalah aktivitas perdagangan yang melibatkan pembelian, penjualan, dan pertukaran barang, jasa, atau informasi melalui jaringan internet. Dalam e-commerce, transaksi bisnis dilakukan secara online melalui platform digital, seperti situs *web*, aplikasi e-commerce, atau media sosial, yang memudahkan konsumen dan penjual berinteraksi tanpa batasan geografis dan waktu

Implementasi *e-commerce* merupakan proses penerapan sistem perdagangan elektronik dalam aktivitas bisnis, yang melibatkan penggunaan teknologi *digital* untuk memfasilitasi transaksi jual beli barang atau jasa secara online. Proses ini mencakup pengembangan *platform digital*, seperti situs *web* atau aplikasi *e-commerce*, pengintegrasian sistem pembayaran elektronik, pengelolaan

inventaris, hingga pengaturan logistik dan pengiriman barang. Implementasi *e-commerce* tidak hanya sekadar membangun toko *online*, tetapi juga membutuhkan strategi pemasaran digital, manajemen data pelanggan, serta sistem keamanan yang andal untuk melindungi data dan transaksi.

e-commerce adalah sistem yang memberikan kemudahan baik pada pengguna maupun konsumen yang akan menikmati pelayanan dalam transaksi jual beli (Muhammad Yusuf et al, Desember 2022).

2.1.4 Baju Adat

Baju adat adalah pakaian tradisional yang mencerminkan identitas budaya, sejarah, dan kearifan lokal suatu daerah atau suku. Biasanya, baju adat dikenakan dalam upacara adat, perayaan budaya, atau acara resmi untuk menunjukkan kebanggaan terhadap warisan leluhur. Setiap daerah memiliki ciri khas tersendiri dalam motif, warna, bahan, dan aksesori yang sarat makna simbolis dan filosofis. Pakaian Adat adalah Media Pembelajaran Karakter dan Identitas. Pembelajaran nilai-nilai filosofis yang terkandung dalam baju adat terbukti mampu memperkuat identitas budaya dan karakter siswa sejak usia dini, menjadi benteng menghadapi *globalisasi* (Kutari et al, Desember 2025).

2.1.5 Web

Web adalah sebagai *platform* untuk menyampaikan informasi, berinteraksi, dan menyediakan berbagai layanan secara *online* melalui jaringan internet. *Web* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia didalamnya yang menggunakan protokol *HTTP (hyper text transfer protocol)* dan dalam mengunakan perangkat lunak yang disebut *browser*. *Web* adalah suatu

media yang bisa digunakan untuk menampung berbagai jenis informasi seperti teks, suara, gambar, dan animasi yang dimana bisa diakses oleh komputer melalui internet. *Web* emerupakan media informasi berbasis jaringan komputer yang bisa diakses kapan saja dan di mana saja dengan biaya yang murah (Firmansyah, Herman Mei,2023).

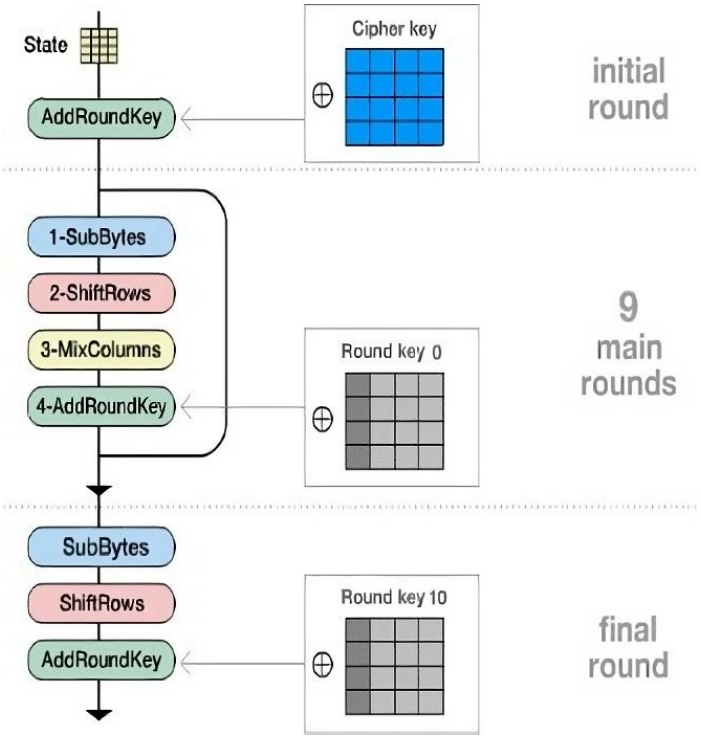
2.1.6 AES (Advanced Encryption Standard)`

Advanced Encryption Standard (AES) merupakan algoritma kriptografi modern yang dapat digunakan untuk mengamankan data. Blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan *dekripsi* (*decipher*) informasi disebut algoritma AES. *Chipertext* adalah data yang tidak dapat dikenal, dan proses mengubah data asli menjadi *chipertext* adalah enkripsi. Sebaliknya *dekripsi* mengubah *ciphertext* data menjadi bentuk asli, yang dikenal sebagai *plaintext*. Algoritma aes memiliki beberapa macam kunci enkripsi dan *dekripsi* pertama 128 bit, kedua 192 bit, dan yang ketiga 256 bit.

Tabel 2. 1 Tabel Putaran Algoritma *AES* tiap blok

Tipe	Kunci	Blok Input	Jumlah n
AES – 128	128-bit	128-bit	10
AES – 192	192-bit	128-bit	12
AES – 256	256-bit	128-bit	14

Pada penelitian ini digunakan AES-128 *bit* yang dilakukan terhadap *state* atau *array of byte* dua dimensi. Sebuah proses penyandian *plaintext* menjadi *ciphertext* adalah penjelasan teknis proses enkripsi atau secara umum enkripsi adalah sebuah proses mengonversikan data normal menjadi data rahasia yang dapat dilihat pada gambar 2. 1.



Gambar 2.1 Proses Enkripsi Algoritma AES-128

Proses enkripsi algoritma AES-128 akan melakukan beberapa Transformasi bytes yaitu AddRoundKey, SubBytes, ShiftRows, dan Mixcolumns yang dipisahkan menjadi 3 bagian, yaitu Initial Round, Main Rounds, dan Final Round.

2.1.7 Pemodelan Berorientasi Objek Menggunakan *Unified Modeling Language (UML)*

Menurut Narulita et al, (Agustus, 2024), UML (*Unified Modeling Language*) membantu pemaparan (deskripsi) dan perancangan sistem, khususnya pada pemrograman berorientasi objek.

UML (*Unified Modeling Language*) merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML juga merupakan Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1) Use Case Diagram

Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi- fungsi tersebut. *Use Case* diagram adalah suatu pola atau gambaran yang menunjukkan perilaku atau kebiasaan sistem (Tabrani et al., 2022).

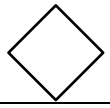
Tabel 2. 2 Simbol - simbol pada *Use Case Diagram*

Simbol	**Notasi**	**Keterangan**
	*Actor*	Aktor bisa berupa pengguna manusia, perangkat keras, atau sistem lainnya.
	*Use Case*	Menggambarkan proses atau tindakan yang dapat dilakukan oleh aktor dalam sistem.
	Assocation	Menghubungkan aktor dengan use case yang menunjukkan interaksi.
	Relasi	menggambarkan alur sistem, fungsionalitas utama, serta skenario tambahan dalam interaksi antara aktor dan sistem
<<include>>	*Include*	menunjukkan bahwa suatu use case selalu menyertakan use case lainnya.
<<extend>>	*Extend*	skenario opsional di mana suatu use case dapat diperluas oleh use case lain dalam kondisi tertentu.

2) Activity Diagram

Menurut novitasari dalam (Kurniawan et al., 2021*) activity diagram* merupakan pemodelan dari sistem untuk menggambarkan aktivitas yang sedang berjalan dalam menjelaskan activitas program tanpa melihat coding yang dibuat.

Tabel 2. 3 Simbol - simbol pada *Activity Diagram*

Simbol	Notasi	Keterangan
	*Initial State*	Menunjukkan titik awal dari sebuah proses atau aktivitas.
	*Final State*	Titik akhir untuk mengakhiri suatu aktivitas
	*Activity*	Merepresentasikan tindakan atau proses yang dilakukan dalam alur kerja.
	*Decision*	Pilihan untuk pengambilan keputusan
	Control Flow	Menunjukkan arah alur aktivitas, menghubungkan satu elemen dengan elemen lainnya.

2.1.8 MYSQL

MySQL adalah sebuah sistem manajemen basis data sederhana yang digunakan untuk mengelola dan mengakses data dalam basis data. Fungsinya adalah untuk melakukan operasi dasar seperti menyimpan, mengubah, mengambil, dan

menghapus data menggunakan perintah *SQL* dengan fokus pada kemudahan dan efisiensi untuk aplikasi kecil atau terdistribusi. *MSQL* sistem manajemen *database SQL* yang bersifat *open source* dan paling populer saat ini.

MySQL adalah solusi yang populer untuk mengelola dan menyimpan data. MySQL adalah sistem manajemen basis data *open-source* yang sangat fleksibel dan aman (Putri et al,2023).

2.1.9 CodeIgniter

CodeIgniter menurut peneliti adalah salah satu *framework PHP* yang ringan, cepat, dan mudah digunakan untuk membangun aplikasi *web*. Dengan fitur seperti arsitektur *MVC (Model-View-Controller)*, *library* bawaan, dan dokumentasi lengkap, *CodeIgniter* membantu pengembang membuat aplikasi yang efisien dan terstruktur tanpa memulai semuanya dari nol.

CodeIgniter adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman *PHP* yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*. Kesimpulan dari pengertian tersebut bahwa *CodeIgniter* adalah *framework PHP* yang di dalamnya terdapat fitur lengkap aplikasi *web* yang sudah dikemas menjadi satu (Nilfaidah et al,2021).

2.1.10 Draw.io

Draw.io menurut peneliti adalah alat *online* yang sederhana dan fleksibel untuk membuat diagram, seperti diagram alur, diagram *ERD*, bagan organisasi, dan lainnya. Dengan antarmuka *drag-and-drop* dan layanan penyimpanan *cloud*, seperti *Google Drive* atau *OneDrive*, *Draw.io* memudahkan pengguna dalam membuat dan menyimpan diagram secara efisien.

Draw.io untuk proses memvisualisasikan konsep-konsep seperti diagram alur kerja, struktur data, dan antarmuka pengguna (Oktaviandi et al, 2024).

2.1.11 Mode Operasi AES CBC (Cipher Block Chaining)

Mode CBC *(Cipher Block Chaining*) bekerja seperti rantai, di mana setiap blok plaintext di-XOR dengan ciphertext sebelumnya—blok pertama memakai IV acak—sehingga hasil enkripsinya selalu berubah-ubah meskipun datanya sama; pola tidak bisa ditebak karena setiap blok saling bergantung, dan perubahan kecil pada satu blok akan merusak blok setelahnya saat didekripsi, sementara proses dekripsi sendiri tinggal membalik langkahnya: mendekripsi tiap blok lalu meng-XOR-kannya dengan ciphertext sebelumnya agar plaintext asli muncul kembali.

Pengembangan algoritma AES-CBC dan implementasinya pada perangkat keras memberikan kontribusi signifikan dalam meningkatkan keamanan komunikasi data, baik dalam lingkungan akademis maupun praktis (P, A, Rizky, et al, 2024).

2.1.12 R&D (*Research and Development*)

Research and Development (R&D) merupakan proses atau langkah langkah untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada. Penelitian pengembangan merupakan salah satu jenis penelitian yang menjadi penghubung atau pemutus kesenjangan antara penelitian dasar dengan peelitian terapan.

Penelitian (R&D) *Research and Development* pengembangan telah menjadi pilihan utama untuk menghasilkan inovasi- inovasi baru di segala aspek bidang kehidupan. Prosedur ilmiah ini menjadi primadona karena memberikan kepastian dalam aktivitas penelitian berupa produk atau model (Marinu Waruwu,2024).

2.2 Penelitian yang Relevan

Penelitian relevan adalah penelitian yang sudah pernah dilakukan sebelumnya dan dianggap relevan atau berkaitan dengan topik penelitian yang akan dilakukan. Penelitian relevan bermanfaat untuk menghindari pengulangan penelitian dengan topik yang sama.

Tabel 2.3 menunjukkan bahwa salah satu manfaat utama dari penelitian relevan adalah membantu peneliti memposisikan penelitiannya dalam konteks yang lebih luas. Dengan mengetahui penelitian-penelitian yang telah dilakukan sebelumnya, peneliti dapat mengidentifikasi celah-celah pengetahuan yang belum terungkap dan merumuskan pertanyaan penelitian yang orisinal.

Selain itu, penelitian relevan juga dapat berfungsi sebagai sumber inspirasi bagi peneliti. Dengan mempelajari metode penelitian yang digunakan oleh peneliti lain, peneliti dapat memperoleh ide-ide baru untuk penelitiannya sendiri.

Tabel 2. 4 Penelitian yang Relevan

No	Penulis/ Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
1	(Ariansyah Putra Ramadani Tarigan, Puji Sari Ramadhan, Kahiri Ibnutama,2023)	Penerapan Kriptografi Untuk Pengamanan Data Penjual Sepatu Dengan Metode AES (Advanced Encryption Standard)	Advanced Encryption Standard (AES)	Enos Ginting adalah salah satu usaha dagang yang menjual sepatu. Karna itu, setiap laporan harus disusun dengan akurat. Dalam komunikasi data terdapat suatu teknik untuk merahasiakan data yang dikenal dengan kriptografi	Hasil penelitian merupakan terciptanya sebuah aplikasi yang dapat mengenkripsi dan *dekripsi* file Excel yang nantinya diharapkan dapat membantu UD.Enos Ginting dalam meningkatkan keamanan data penjualan.
2	(Alfin Gimnastiar, Lailan Sofinah Harahap, Riski Abdillah,2025)	Implementasi *Watermarking* Pada Gambar Menggunakan Matlab Untuk Mencegah Plagiarisme Pada Praktikum	Pendekatan Eksperimental	terutama dalam laporan praktikum mahasiswa. Untuk mencegah hal tersebut, penelitian ini mengimplementasikan teknik *Watermarking* teks secara langsung pada gambar digital menggunakan fungsi insertText() pada perangkat lunak MATLAB.	Proses *Watermarking* dilakukan dengan menyisipkan teks ke dalam gambar menggunakan fungsi program pada MATLAB. Teks yang disisipkan berfungsi sebagai identitas pemilik gambar, dalam penelitianini adalah nama mahasiswa

No	Penulis/Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
3	(Ilham wahyu Kuncoro Aji, Reva Ragam Santika,2024)	Penerapan Kriptografi Menggunakan Metode AES Untuk Pengamanan Data Penjualan Rumah Makan Mitra Minang	Advanced Encryption Standard (AES)	Penerapan AES pada sistem transaksi Rumah Makan Mitra Minang melibatkan data transaksi pembayaran, yang mencakup informasi pribadi pelanggan, serta data transaksi lainnya.	Pada hasil pengujian sistem yang dibuat menunjukkan bahwa pada penerapan metode AES secara signifikan meningkatkan keamanan data transaksi dengan mengurangi potensi risiko pencurian data dari serangan siber maupun dari pihak yang tidak bertanggung jawab.
4	(Muhammad Wisnu Ramadhana, abdullah Muhazir, Jaka Prayudha,2024)	Pengamanan Data Penjualan Biji Kopi Sembekandua Menggunakan Metode Advanced Encryption Standard	Advanced Encryption Standard (AES)	Permasalahan yang ada di CV.Muda Kopi yaitu arsip data belum memiliki sistem keamanan, sehingga memungkinkan adanya pihak yang tidak bertanggung jawab untuk memodifikasi dan melakukan pencurian terhadap data-data tersebut.	Berdasarkan hasil analisa, dalam penyelesaian masalah pengamanan data penjualan biji kopi sembekandua pada CV.Muda Kopi, algoritma advanced encryption standard 128-bit berhasil diterapkan. Kebutuhan pada sistem telah sesuai dengan kebutuhan dalam pengamanan data penjualan biji kopi sembekandua pada CV.Muda Kopi.

No	Penulis/Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
5	(Leonardi Sidabutar, Mukhlis Ramadhan, zaimah Panjaitan,2024)	Implementasi Kriptogrtafi Pengamanan Data Pemesanan Produk Menggunakan Meotode AES	Advanced Encryption Standard (AES)	Kriptografi merupakan seni dan ilmu untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditunjukan untuk orang yang hanya memiliki sebuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan.	Algoritma Advanced Encryption Standard 128 dapat digunakan dalam melakukan pengamanan data pemesanan produk pada PT. Sentral Sehat Sejahtera dengan melakukan enkripsi setiap data pemesanan produk yang diinput oleh customer menjadi bentukk kode-kode yang tidak dapat diketahui lagi artinya dengan menggunakan algoritma perhitungan metode AES 128. sistem keamanan data pemesanan produk PT. Sentral Sehat Sejahtera dirancang dengan menentukan use case, activity diagram dan class diagram terlebih dahulu. Dengan demikian akan didapatkan alur cara kerja sistem yang akan dibuat sehingga dapat dirancang struktur database dan antarmuka sistem.

Perbedaan Penelitian yang diambil oleh penulis dan referensi dari contoh penelitian yang relevan adalah semua referensi hanya memakai satu keamanan misalnya hanya menggunakan AES dan tidak menggunakan *Watermarking* begitu juga sebaliknya, sedangkan yang penulis buat pada tugas akhir ini memakai dua keamanan sekaligus yaitu AES dan *Watermarking*.

BAB III

METODOLOGI PENELITIAN

3.1 Tinjauan Umum

Dengan bergesernya objek-objek hak cipta ke bentuk digital telah mengubah cara penyebaran dan pemanfaatannya melalui media internet. Begitu pula, perangkat lunak kini tidak memerlukan media CD (*compact disc*) untuk proses instalasi. Objek digital, yang merupakan barang elektronik tanpa wujud fisik, dapat ditemukan dalam berbagai media digital.

Gambar baju adat yang diunggah tanpa perlindungan dapat dengan mudah disalin, dimanipulasi, atau disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab misalnya (*reaploader* ke *marketplace* lain). Selain itu, data transaksi dan informasi pengguna juga rentan terhadap ancaman keamanan jika tidak dienkripsi dengan baik. Oleh karena itu, diperlukan upaya untuk mengamankan data dan konten digital tersebut guna menjaga keaslian, integritas, dan kerahasiaan informasi dalam sistem penjualan online.

Tugas akhir ini membahas implementasi sistem keamanan pada aplikasi penjualan baju adat berbasis *web* dengan memadukan dua teknologi utama, yaitu digital *Watermarking* dan *Enkripsi AES (Advanced Encryption Standard). Watermarking* digunakan untuk melindungi gambar produk dari pembajakan dengan menyisipkan identitas tersembunyi ke dalam gambar, tanpa mengganggu tampilan visualnya. Sementara itu, *Enkripsi AES* digunakan untuk mengamankan data sensitif seperti informasi pribadi pengguna dan detail transaksi agar tidak mudah diakses oleh pihak yang tidak berwenang.

3.2 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah *Research and Development* (R&D) karena metode penelitian R&D ini adalah jenis penelitian yang tahapannya meliputi penelitian awal untuk memahami masalah, pengembangan solusi atau produk, pengujian, serta penyempurnaan hingga produk siap digunakan atau diterapkan.

3.3 Desain Penelitian

Pada penelitian ini penulis menggunakan desain penelitian *Research and Development* (R&D). Dan berikut tahapan yang peneliti lakukan antara lain :

a. Analisis kebutuhan

Analisis kebutuhan dilakukan untuk merancang dan membangun sistem penjualan baju adat berbasis *web* yang aman dan andal dengan fitur proteksi gambar *(Watermarking)* serta perlindungan data pengguna *(Enkripsi AES).*

b. Design Sistem

Tahap ini mencakup perancangan arsitektur sistem, desain *database*, dan antarmuka pengguna. Proses *implementasi* menggunakan *Watermaking* dan *Enkripsi AES* dirancang untuk menjamin keamanan saat bertransaksi.

c. *Implementasi* Sistem

Implementasi dilakukan dengan membangun sistem berbasis *web* menggunakan bahasa pemrograman *PHP* dan *framework Codeignier*. *Database* menggunakan *MySQL* untuk menyimpan data *implementasi* penjualan, sementara proses *Enkripsi AES* diintegrasikan untuk melindungi data.

d. Pengujian Sistem

Pengujian dilakukan untuk memastikan sistem berfungsi sesuai dengan kebutuhan. Metode pengujian yang digunakan adalah *Black Box Testing*, yang difokuskan pada pengujian fungsi-fungsi sistem.

e. Evaluasi Sistem

Evaluasi sistem dilakukan melalui penyusunan laporan yang mencakup hasil pengujian performa dan keamanan sistem secara menyeluruh, dengan tujuan untuk menilai efektivitas *implementasi* fitur-fitur utama, seperti kecepatan proses, serta tingkat perlindungan terhadap potensi ancaman keamanan, sehingga dapat memberikan gambaran menyeluruh mengenai keandalan dan kelayakan sistem yang telah dirancang.

3.4 Teknik Pengumpuan Data

Teknik pengumpulan data dalam kegiatan penelitian ini dilakukan dengan studi pustaka, observasi, dan wawancara yaitu :

a. Studi Pustaka

Penelitian yang dilakukan dengan cara membaca dan mempelajari buku-buku atau referensi dari internet,dan lainnya yang berhubungan dengan masalah yang akan di teliti sebagai tujuan pustaka.

b. Observasi

Pengumpulan data dengan melakukan pengamatan langsung kelapangan untuk memperoleh data yang lengkap dan akurat untuk kebutuhan implementasi penjualan yang akan dibangun.

c. Wawancara

Pengumpulan data dengan cara melakukan komunikasi dan wawancara secara langsung dengan pihak-pihak terkait yaitu, pemilik took baju adat bawi lamus

3.5 Analisis Kebutuhan

Adapun dalam pengembangan sistem ini, agar sistem yang dibangundapat berjalan dengan baik maka dibutuhkan suatu analis sebagai berikut. Berikut kebutuhan perangkat lunak dapat di lihat pada Tabel 3.1

Tabel 3.1 Kebutuhan Perangkat Lunak

Perangkat Lunak	Spesifikasi
Windows 11 Home 64-Bit	Sebagai sistem operasi
PHP versi 8 dan HTML versi 5	Sebagai bahasa pemrograman
CodeIgniter 4 dan Bootstrap 4	Sebagai Framework sistem
Xampp versi 3.2.4	Sebagai *web server* lokal / *localhost*
Mysql	Sebagai database
Visual studio code	Sebagai kode editor untuk proses pembuatan sistem informasi
Canva	Sebagai perancang desain antarmuka dan alat
Draw.io	Sebagai perancang flowchart

Pada penelitian ini penulis juga menggunakan perangkat keras computer dalam pembuatan sistem.

Berikut kebutuhan perangkat keras dapat dilihat pada Tabel 3.2

Tabel 3.2 Kebutuhan Perangkat Keras

Perangkat Keras	Spesifikasi
Type	AVITA NE14A2
Processor	Intel® Celeron® N4020 CPU @ 1.10GHZ (2 CPUs), ~1.1GHz
Memory	4,00 GB

3.5.1 Analisis Kebutuhan Perangkat keras dan Perangkat Lunak Untuk Menjalankan Sistem

Adapun kebutuhan perangkat keras dan perangkat lunak untuk menjalankan sistem sebagai berikut:

a. Analisis Kebutuhan Perangkat Keras

Perangkat keras *(hardware)* yang digunakan dalam pembuatan sistem informasi ini adalah *komputer, laptop, tablet*, dan *handphone* yang sudah bisa menggunakan *browser* serta *internet*. Untuk spesifikasi masing – masing perangkat keras tidak memiliki spesifikasi khusus dikarenakan sistem berupa *website* yang dijalankan diperangkat yang memiliki *browser* dan *internet*.

b. Analisis Perangkat Lunak

Perangkat lunak *(software)* yang digunakan untuk mengakses sistem informasi ini adalah sistem operasi *Windows, Ios, MacOs, Android*, dan *Linux.* Untuk *browsing internet* bisa menggunakan *Google Chrome, Microsoft Edge, Safari, Mozila, Firefox*, ataupun *browser* lain yang bisa mengindeks sistem informasi ini di mesin pencari.

3.6 Desain Sistem

Pada bagian ini, akan diuraikan secara jelas rancangan dan desain yang bertujuan untuk menggambarkan struktur sistem yang akan bertujuan untuk menggambarkan struktur sistem yang akan dibuat oleh penulis.

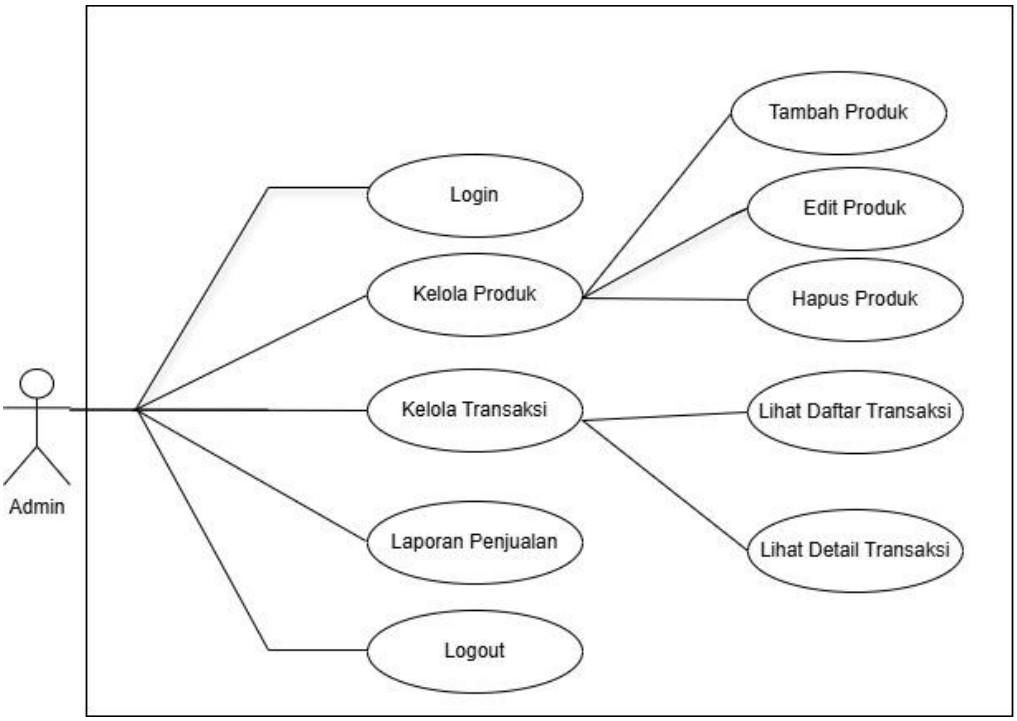
3.6.1 Desain Proses

Dalam desain sistem, disini penulis menggunakan diagram UML (*Unified Modeling Languange)* dan *Flowchart.* Adapun diagram yang digunakan adalah *Use Case Diagram, Activity diagram* juga *Flowchart*.

a. Use Case Diagram

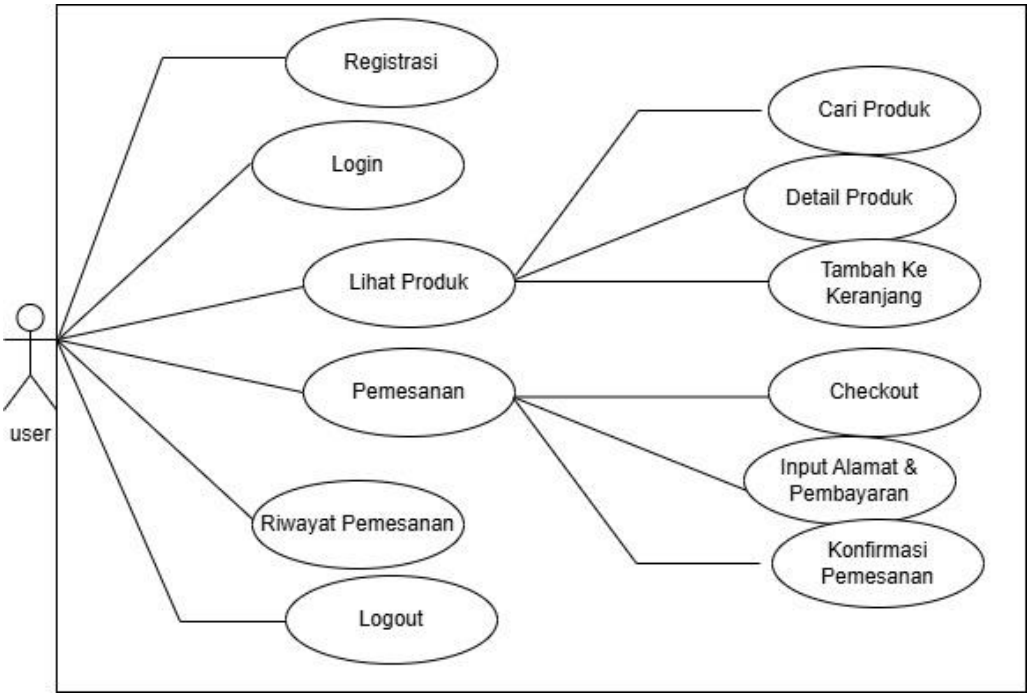
Use Case Diagram adalah salah satu jenis *diagram* dalam pemodelan perangkat lunak yang digunakan untuk menggambarkan interaksi antara sistem perangkat lunak dengan berbagai aktor atau entitas yang terlibat. Tujuannya adalah untuk mengidentifikasi, menggambarkan, dan memahami bagaimana pengguna atau aktor eksternal berinteraksi dengan sistem perangkat lunak, sertamenentukan fungsionalitas utama yang akan disediakan oleh sistem.

Gambar 3. 1 merupakan *use case diagram* dari user administrator, menu yang dapat diakses, serta proses yang dialami oleh admin ketika aplikasi dijalankan.



Gambar 3. 1 *Use Case Diagram Admin*

Gambar 3. 2 merupakan *use case diagram* dari user umum, menu yang dapat diakses, serta proses yang dialami oleh umum ketika aplikasi dijalankan.



Gambar 3. 2 *Use Case Diagram Umum*

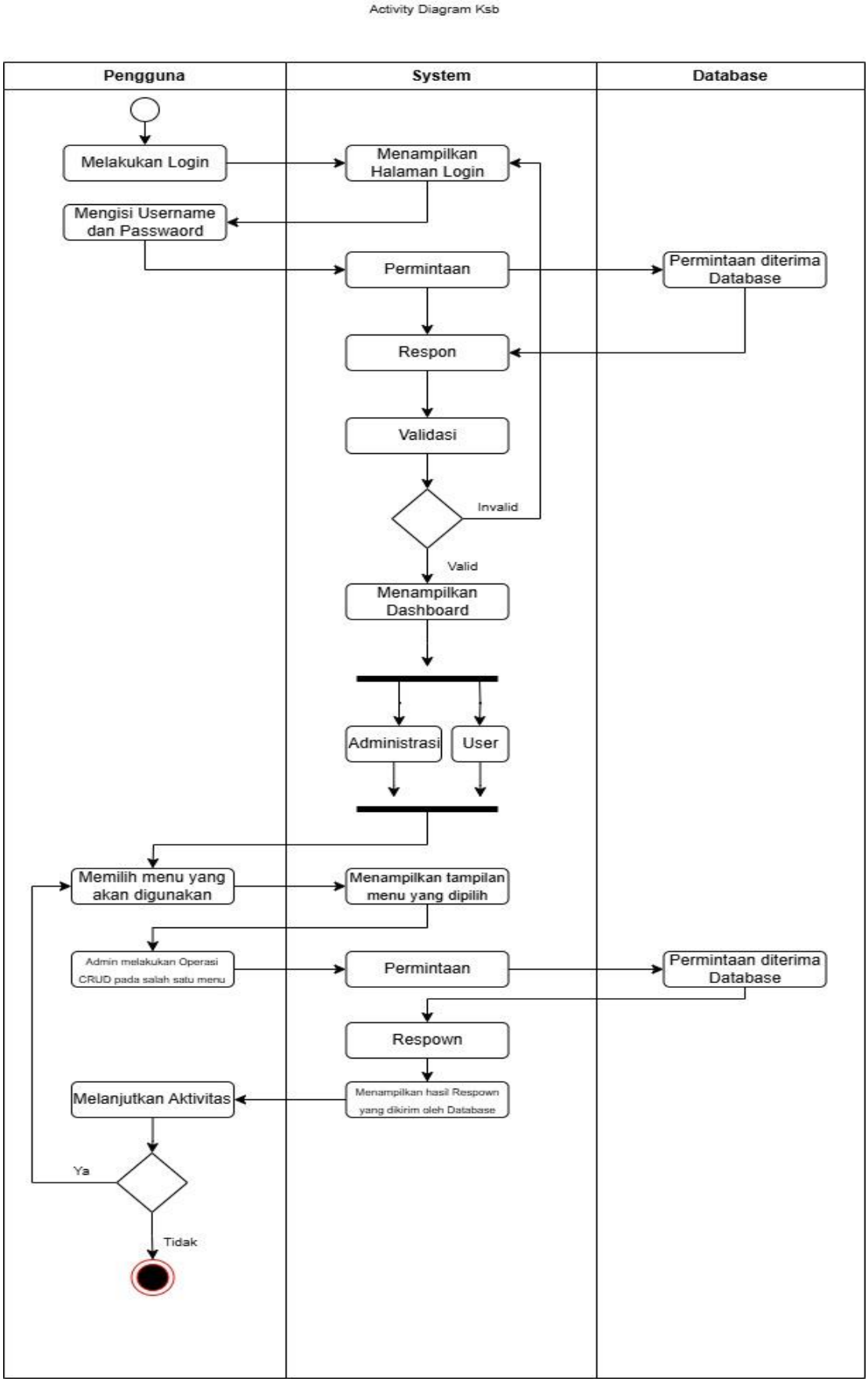
b. Activity Diagram

Pada pemodelan UML, diagram aktivitas (*activity diagram)* adalah jenis diagram dalam pemodelan perangkat lunak yang digunakan untuk menggambarkan alur kerja atau aktivitas yang terjadi dalam suatu proses atau sistem. Tujuannya adalah untuk memberikan pandangan visual tentang bagaimana tugas, aktivitas, dan perubahan satus berlangsung dalam suatu prosesatau interaksi.

a) Activity Diagram Sistem

Gambar 3. 3 merupakan *activity diagram* yang menggambarkan alur proses *login* hingga pengelolaan data dalam sebuah sistem. Dimulai dengan petugas yang melakukan login, sistem menampilkan *form login* untuk diisi dengan *username* dan *password*. Data yang dimasukkan akan dikirim sebagai *request* ke *database* untuk diproses. Setelah mendapatkan

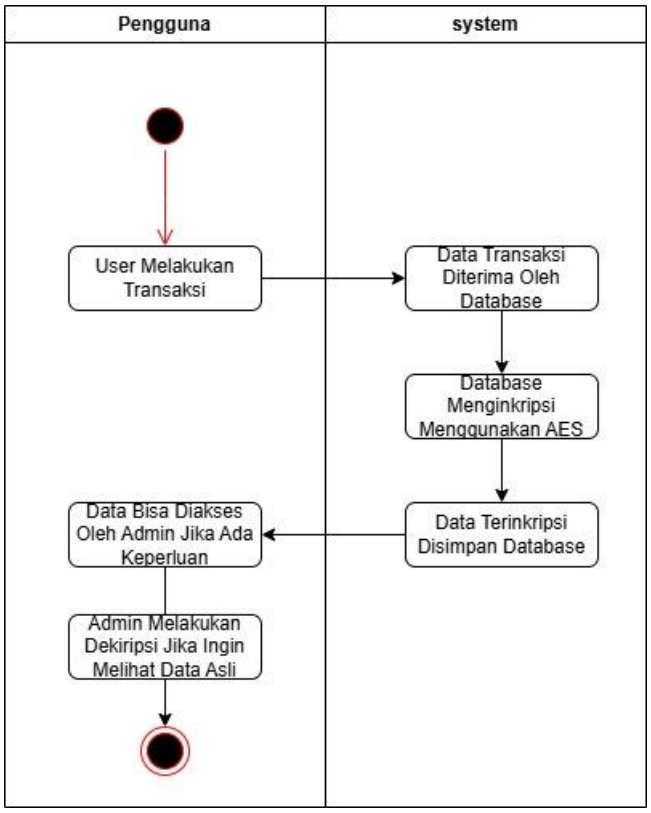
respon dari *database*, sistem melakukan validasi. Jika validasi berhasil, *dashboard* akan ditampilkan sesuai kategori (Administrasi atau Umum). Petugas dapat memilih menu yang diinginkan, kemudian sistem menampilkan tampilan menu tersebut. Selanjutnya, petugas dapat melakukan operasi CRUD *(Create, Read, Update, Delete)* pada menu tersebut, yang melibatkan pengiriman *request* ke *database* untuk diproses dan menerima respon dari *database*. Hasil respon akan ditampilkan pada sistem, dan petugas dapat memutuskan untuk melanjutkan aktivitas atau mengakhiri proses.



Gambar 3. 3 *Activity Diagram Sistem*

b) Activity Diagram Enkripsi

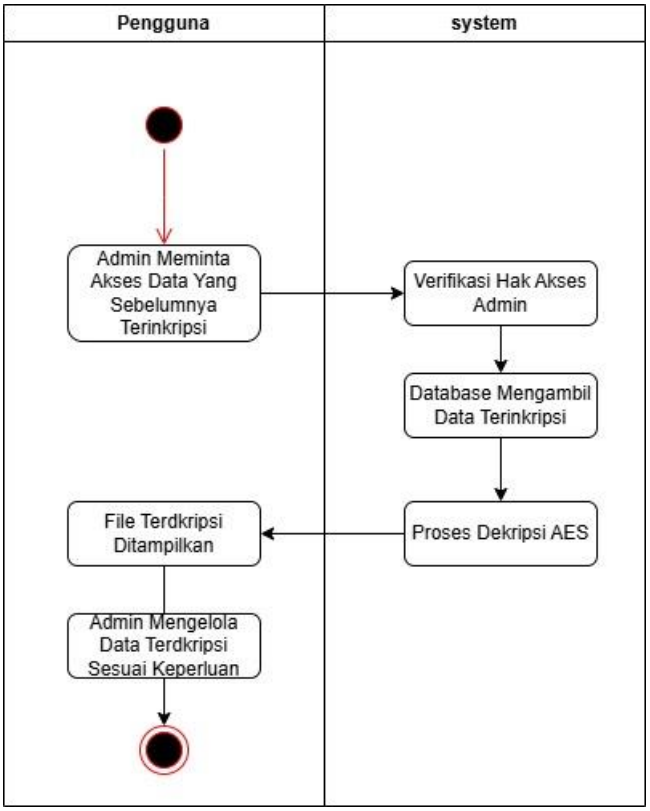
Gambar 3.4 Activity diagram ini menggambarkan proses alur *enkripsi* data transaksi pada sistem penjualan , dimulai dari pengguna yang melakukan transaksi. Data transaksi kemudian diterima oleh sistem dan langsung diproses oleh *database* yang melakukan *enkripsi* menggunakan algoritma *AES*. Setelah terenkripsi, data disimpan secara aman di dalam *database*. Jika ada keperluan, *admin* yang memiliki *otorisasi* dapat mengakses data tersebut. Untuk melihat isi data aslinya, *admin* harus melakukan proses *dekripsi* terlebih dahulu. Alur ini menunjukkan bagaimana sistem menjaga keamanan data dengan enkripsi langsung oleh database.



Gambar 3.4 *Activity Diagram Emkripsi*

c) Activity Diagram Dekripsi

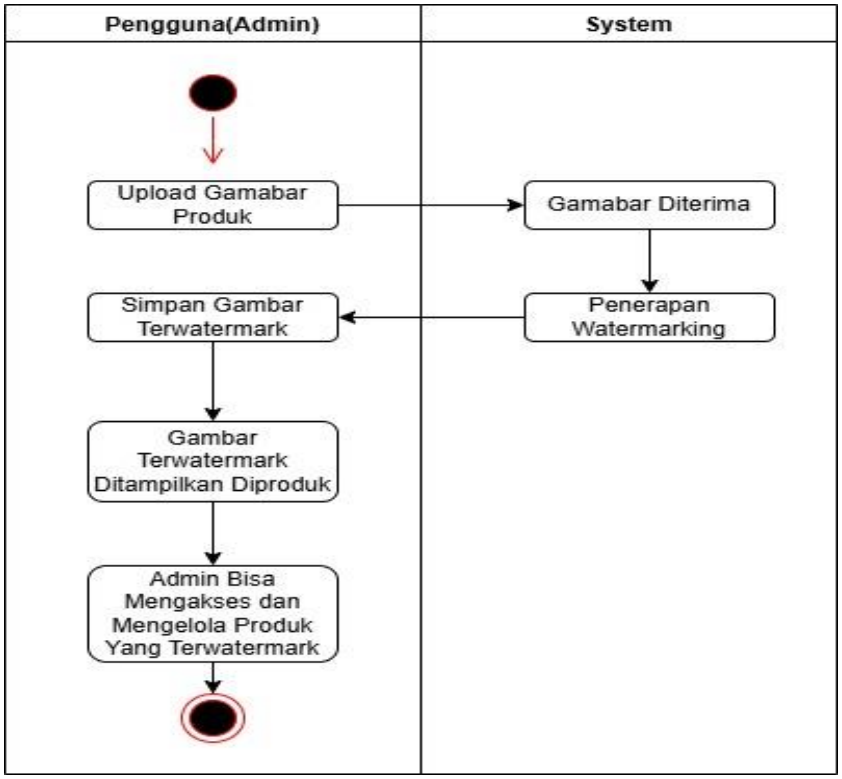
Gambar 3.5 merupakan alur pada *activity diagram dekripsi* data oleh *admin* pada sistem *e-commerce* ini dimulai dari *admin* yang *login* ke dalam sistem sebagai langkah awal untuk memperoleh hak akses. Setelah berhasil *login*, *admin* kemudian meminta akses terhadap data transaksi yang telah *terenkripsi* di *database*. Sistem akan melakukan *verifikasi* terhadap hak akses *admin* untuk memastikan bahwa permintaan tersebut sah dan sesuai *otorisasi*. Jika terverifikasi, *database* mengambil data terenkripsi, lalu melakukan proses *dekripsi* menggunakan algoritma *AES* agar data dapat dikembalikan ke bentuk aslinya. Setelah berhasil di*dekripsi*, data asli ditampilkan kepada *admin*. Pada tahap akhir, *admin* dapat mengelola atau menggunakan data tersebut sesuai dengan keperluan, seperti untuk analisis, laporan, atau *validasi* transaksi. Proses ini memastikan bahwa hanya pihak yang berwenang yang dapat melihat isi data yang dilindungi, menjaga keamanan dan kerahasiaan transaksi.



Gambar 3.5 *Activity Diagram Dekripsi*

d) Activity Diagram Watermarking

Gambar 3.6 proses *Watermarking* gambar produk pada *web* penjualan baju adat dimulai ketika *admin login* ke dalam sistem untuk mengelola produk. Setelah berhasil masuk, *admin* mengunggah gambar produk yang kemudian diterima oleh *server*. *Server* memproses gambar tersebut dengan menerapkan algoritma *Watermarking*, secara terlihat *(visible)* untuk melindungi hak cipta dan identitas toko. Setelah proses penyisipan *watermark* selesai, gambar disimpan kembali di sistem dan ditampilkan di halaman *web*. Pengguna yang mengakses situs akan melihat gambar produk yang telah diberi *watermark*, sehingga gambar tersebut aman dari pencurian atau penyalahgunaan. Proses ini memastikan keamanan *visual* dan *digital* dari aset gambar dalam *platform e-commerce*.



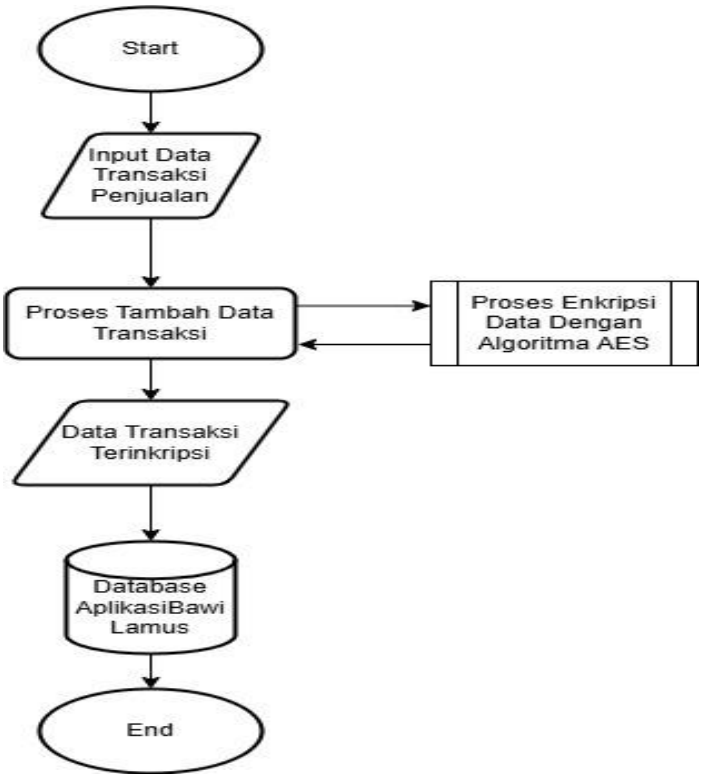
Gambar 3.6 *Activity Diagram Watermarking*

c. Flowchart

Pada sistem keamanan aplikasi penjualan baju adat Bawi Lamus Palangkaraya, algoritma *AES*-128 diimplementasikan pada proses penambahan data, baik untuk data transaksi penjualan maupun data pengguna. File dokumen seperti bukti pembayaran atau laporan yang diunggah akan dienkripsi terlebih dahulu menggunakan algoritma *AES* sebelum disimpan ke dalam *database*. Proses dimulai ketika petugas memilih file yang ingin diamankan, kemudian memasukkan data yang diperlukan. Setelah itu, sistem melakukan enkripsi terhadap file menggunakan metode *AES_Encrypt* dan hasilnya disimpan dalam bentuk terenkripsi di *database*. Dengan cara ini, file dokumen penting menjadi lebih aman dari ancaman akses tidak sah dan serangan seperti *SQL Injection,* sekaligus menjaga kerahasiaan dan integritas data dalam aplikasi:

a) *Flowchart enkripsi* data file Transaksi

Gambar 3. 7 merupakan gambaran alur dari data Transaksi yang ditambahkan oleh *admin* kemudian dienkripsi lalu disimpan kedalam *database*.



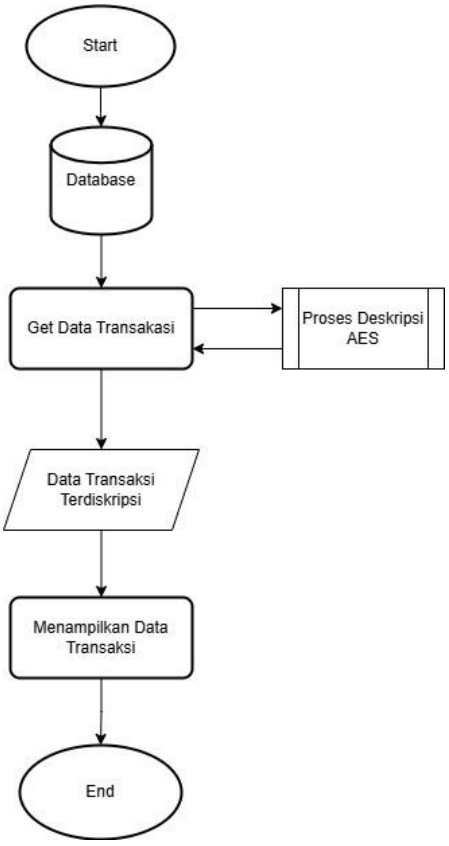
Gambar 3. 7 *Flowchart Enkripsi* Data Transaksi

Selain alur dari proses *enkripsi* data transaksi dan informasi pengguna, juga terdapat alur proses *dekripsi* data dalam penjualan baju adat. Proses ini dimulai dengan pengambilan data terenkripsi dari *database*, yang kemudian diarahkan ke tahapan *Get Data Penjualan*. Selanjutnya, sistem melakukan proses *dekripsi* menggunakan algoritma *AES* untuk mengembalikan data ke bentuk aslinya. Hasil dari proses *dekripsi* ini berupa informasi pembelian, nomor pembeli, alamat pembeli. Data yang telah di*dekripsi* tersebut kemudian ditampilkan dalam antarmuka aplikasi untuk kebutuhan administrasi maupun

analisis. Proses ini memastikan bahwa meskipun data disimpan dalam bentuk terenkripsi, sistem tetap mampu menyajikan informasi secara aman dan utuh. Agar lebih jelas, berikut ditampilkan *flowchart* dari alur proses *dekripsi* data penjualan baju adat.

b) *Flowchart* proses *dekripsi* data penjualan

Gambar 3. 8 alur dari proses *dekripsi* data penjualan kemudian ditampilkan ke *interface* ketika *menu* administrasi di akses.



Gambar 3. 8 .*Flowchart Dekripsi* Data Transaksi

3.6.2 Desain *Enkripsi AES*

Enkripsi Advanced Encryption Standard (AES) merupakan algoritma kriptografi simetris yang banyak digunakan karena tingkat keamanannya yang tinggi serta efisiensinya dalam proses enkripsi dan *dekripsi* data. Dalam konteks Implementasi Keamanan Berbasis Web Menggunakan Watermarking dan Enkripsi AES, penulis menggunakan mode operasi AES CBC *(Cipher Block Chaining)* Mode kerja *CBC* dalam sistem berlangsung dengan meng-XOR setiap blok data dengan ciphertext sebelumnya—blok awal memakai IV acak—sebelum dienkripsi, sehingga hasil enkripsi selalu berubah dan tidak menampilkan pola asli. Dalam implementasinya, ketika admin mengenkripsi, sistem membagi data menjadi blok-blok, menggabungkannya dengan IV dan ciphertext sebelumnya, lalu menghasilkan ciphertext berantai yang aman; sedangkan saat admin mendekripsi, sistem membalik proses tersebut dengan mendekripsi tiap blok lalu meng-XOR-nya kembali dengan blok ciphertext sebelumnya atau IV, sehingga data asli dapat dipulihkan dengan tepat.

a. .Desain Arsitektur Sistem

   a) *Frontend* Aplikasi berbasis *web* menggunakan *framework CodeIgniter* 4 yang menampilkan halaman toko, *login* pengguna, dan formulir transaksi.

   b) *Backend*: *API* dibangun menggunakan *Node.js* atau *PHP*, yang bertugas mengelola proses enkripsi dan *dekripsi* data sensitif dalam transaksi penjualan.

   c) *Database: MySQL* digunakan sebagai media penyimpanan utama untuk data transaksi dan akun pengguna, dalam bentuk terenkripsi.

b. Data yang Akan Dienkripsi

Berikut adalah data sensitif yang akan diamankan menggunakan AES:

a) Informasi Pembeli:

1) Tujuan: Menjaga identitas pengguna agar tidak disalahgunakan.

2) Format: Alamat, nomor kontak. .

c. Proses Kerja AES pada Aplikasi

a) Penyimpanan Data

1) Input Data: Pengguna mengisi form pembelian atau registrasi melalui antarmuka *web.*

2) Enkripsi: Backend menginkripsi data sensitive menggunakan algoritma AES dengan kunci rahasia milik sistem.

3) Penyimpanan: Data yang sudah terenkripsi disimpan dalam *database MySQL* atau *storage* file *server*.

b) Pengambilan Data

1) Query: Sistem melakukan pengambilan data transaksi atau akun yang telah terenkripsi dari database.

2) *Dekripsi*: Data dideskripsi menggunakan kunci rahasia untuk ditampilkan secara aman.

3) Output: Informasi ditampilkan kembali kepada pengguna yang berhak dalam format asli (plaintext) untuk keperluan seperti cek pesanan, konfirmasi transaksi, atau pengelolaan produk.

d. Struktur Database yang Terenkripsi

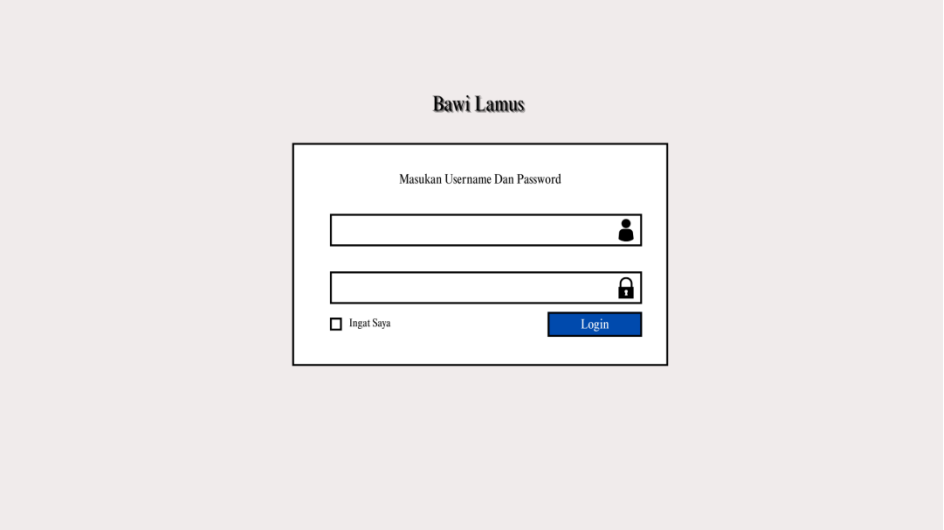
Tabel 3.3 Tabel *Database* yang *Terenkripsi*

Tabel	Kolom	Enkripsi	Keterangan
Pengguna	Alamat	AES-128	Alamat dienkripsi untuk menjaga kerahasian data pelanggan
	No_Telepon	AES-128	No Telepon dienkripsi untuk menjaga data pelanggan
Transaksi	Alamat_Pengiriman	AES-128	Alamat Pengiriman dienkripsi untuk menjaga data pelanggan

3.6.3 Desain *Interface*

Pada tahap ini akan dijelaskan desain *interface* pada halaman – halaman didalam media pembelajaran ini nantinya. Adapun rancangan desain *interface* adalah sebagai berikut:

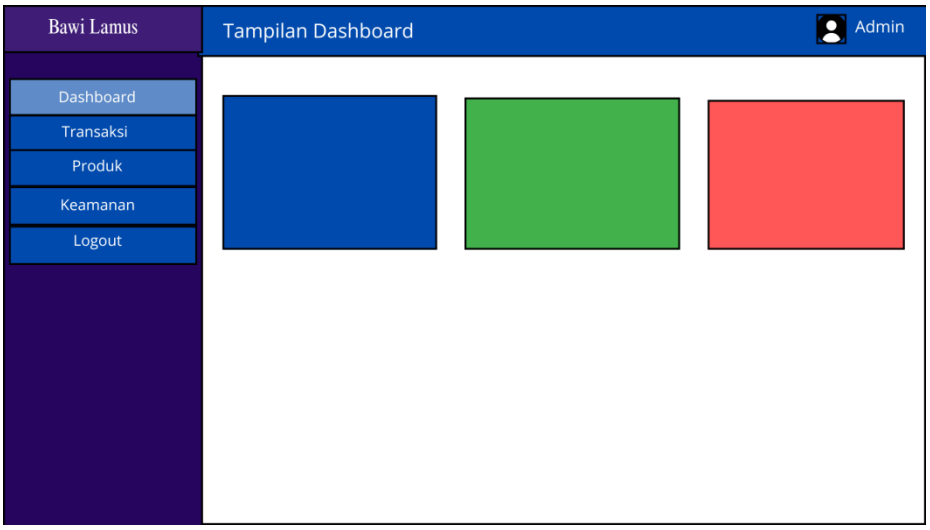
a. Desain Halaman *Login*



Gambar 3. 9 Desain Halaman *Login*

b. Desain Halaman *Dashboard*

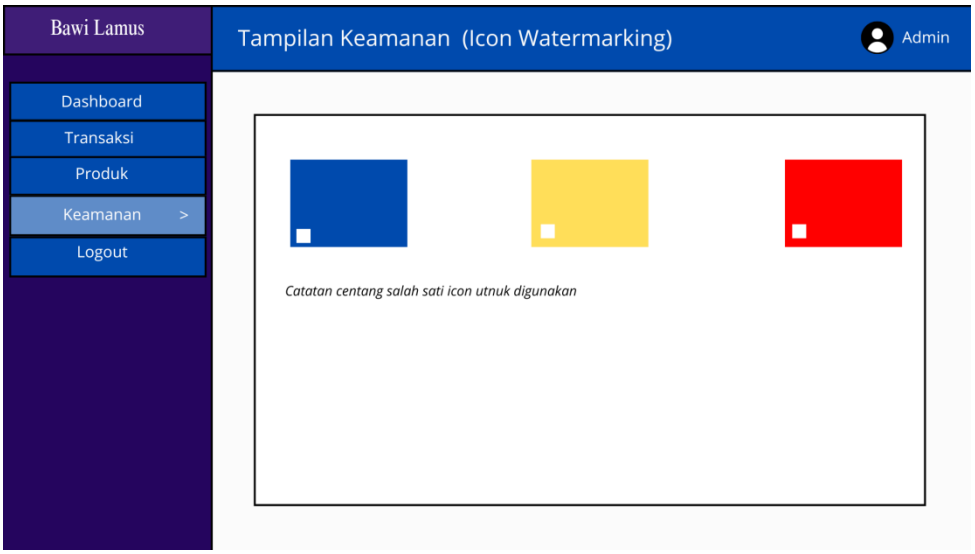
Tampilan Desain *dasboard* adalah rancangan yang akan ditampilkan ketika admin atau operator berhasil *login*. Halaman ini berisikan Dashboard untuk ringkasan fitur, transaksi, produk, keamanan, dan keluar/*logout*.Tampilan halaman *dasboard* dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Desain Halaman *dashboard*

c. Desain Proses *Watermarking*

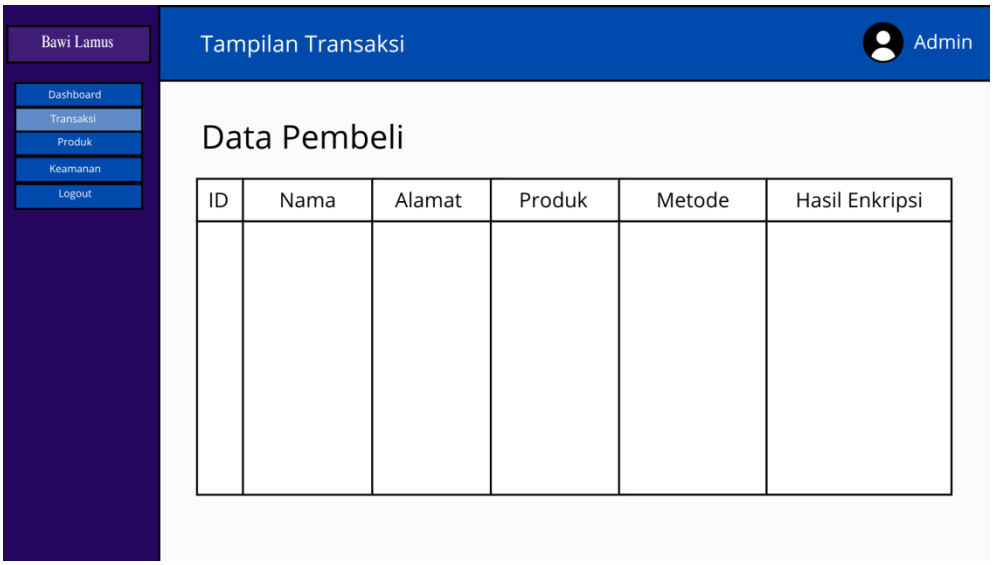
Proses *Watermarking* dimulai ketika *admin* masuk ke sistem dan mengakses fitur keamanan melalui *dashboard*. Di dalam fitur keamanan, *admin* memilih opsi *Watermarking*, lalu sistem menampilkan beberapa pilihan gambar *ikon* sebagai desain *watermark*. *Admin* kemudian diminta untuk mencentang salah satu gambar ikon yang akan digunakan sebagai *watermark* tetap. Setelah memilih dan menyimpan pengaturan tersebut, setiap kali admin mengunggah gambar produk baru, sistem secara otomatis menyisipkan *watermark* sesuai dengan ikon yang telah dipilih sebelumnya. Gambar produk yang telah diberi *watermark* akan tersimpan dan ditampilkan kepada pengguna dengan perlindungan *visual* tersebut. Tampilan desain dapat dilihat pada gambar 3.11



Gambar 3. 11 Desain Proses *Watermarking*

d. Desain Data Transaksi

Proses data transaksi dimulai ketika pembeli melakukan submit pada formulir pembelian, di mana data yang dikirim meliputi ID, nama, alamat, produk yang dibeli, dan metode pembayaran. Setelah data terkirim, sistem akan langsung melakukan proses enkripsi terhadap informasi penting, seperti nama dan alamat, guna menjaga keamanan data pribadi pembeli. Hasil dari proses ini kemudian ditampilkan dalam bentuk tabel pada halaman transaksi admin, dengan kolom yang terdiri dari ID, Nama, Alamat, Produk, Metode, dan Hasil Enkripsi. Tampilan ini memungkinkan admin untuk memantau transaksi sekaligus memastikan bahwa data yang tersimpan telah melalui perlindungan enkripsi..Tampilan desain bisa dilihat pada Gambar 3.12

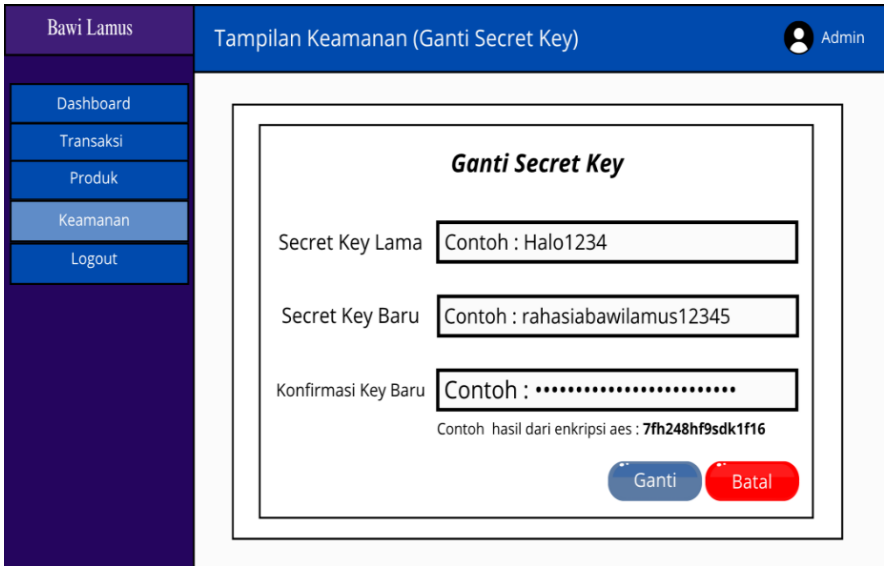


Gambar 3. 12 Desain Data Transaksi

e. Desain *Secret Key*

Proses dimulai ketika *admin* memasuki fitur keamanan dan memilih opsi "Ganti *Secret Key*", dimana *admin* diberikan akses untuk

mengganti kunci rahasia sistem. *Admin* diminta memasukkan *secret key* lama sebagai verifikasi, lalu membuat kunci baru dan mengisinya kembali pada kolom konfirmasi untuk memastikan kecocokan. Setelah semua data diinput dengan benar dan tombol "Ganti" ditekan, sistem akan memvalidasi dan menyimpan kunci baru tersebut. Jika berhasil, kunci baru tersebut siap digunakan untuk proses enkripsi demi menjaga keamanan data dalam sistem. Tampilan desain bisa dilihat pada Gambar 3.13



Gambar 3.13 Desain Secret Key

f. Desain *Enkripsi Database*

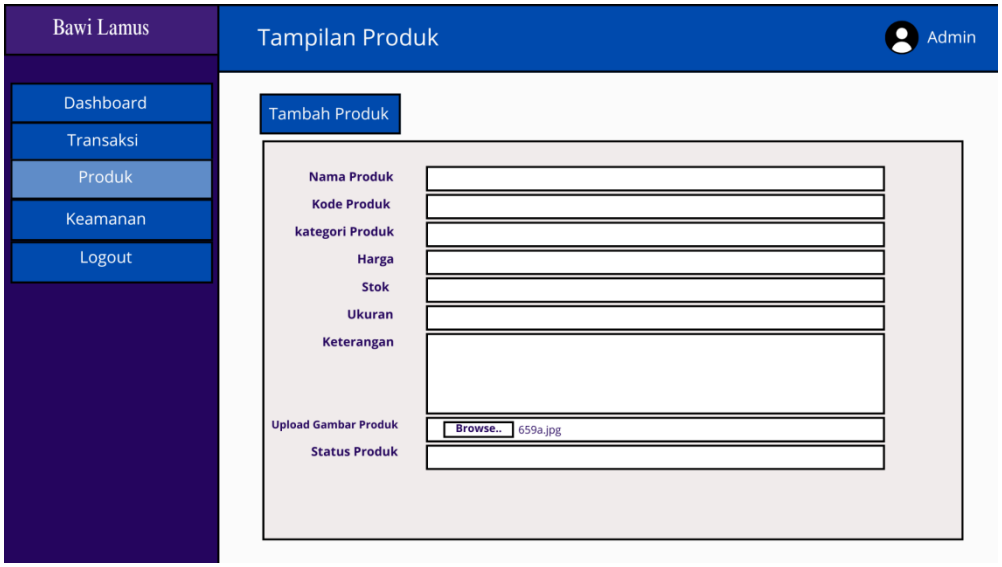
Ini adalah tampilan desain data transaksi yang terinkripsi di database menggunakan *AES* 128. Tampilan desain bisa dilihat pada Gambar 3.14



Gambar 3.14 Desain *Enkripsi Database*

g. Desain Tambah Produk

Dimulai ketika *admin* akses fitur Produk maka *admin* bisa mengakses tambah produk dimana pada desain ini admin akan mengapload produk apa yaang akan dijual, setelah berhasil di upload tambah produk oleh *admin* produk langsung ditampilkan kehalaman pembeli. Tampilan desain bisa dilihat pada Gambar 3.15



Gambar 3.15 Desain Tambah Produk

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Gambaran Umum Sistem

Sistem implementasi keamanan penjualan baju adat berbasis *web* menggunakan *Watermarking* dan *Enkripsi AES* dirancang untuk melindungi hak cipta serta keamanan data dalam transaksi *online*. Dalam sistem ini, teknologi *Watermarking* digunakan untuk menyisipkan tanda digital pada gambar produk baju adat sehingga mencegah pencurian atau penyalahgunaan gambar oleh pihak lain, sementara *Enkripsi AES (Advanced Encryption Standard)* diterapkan untuk mengamankan data sensitif seperti informasi pengguna dan detail transaksi agar tidak dapat diakses oleh pihak yang tidak berwenang. Dengan integrasi kedua teknologi ini, sistem tidak hanya memastikan keaslian dan keutuhan gambar produk, tetapi juga menjamin kerahasiaan dan keamanan data pelanggan, sehingga meningkatkan kepercayaan pengguna terhadap platform penjualan baju adat secara digital.

4.1.2 Proses *Enkripsi AES*-128

```
/**
 * Enkripsi data dengan password
 */
public function encrypt(string $plaintext, string $password): array
{
    $this->validateEncryptionInput($plaintext, $password);

    $iv = $this->generateRandomBytes($this->ivLength);
    $salt = $this->generateRandomBytes(self::SALT_LENGTH);
    $key = $this->deriveKey($password, $salt);

    $ciphertext = openssl_encrypt(
        $plaintext,
        self::CIPHER,
        $key,
        OPENSSL_RAW_DATA,
```

```
            $iv
        );

        if ($ciphertext === false) {
            throw new RuntimeException('Encryption failed: ' . openssl_error_string());
        }

        return [
            'ciphertext' => base64_encode($ciphertext),
            'iv' => base64_encode($iv),
            'salt' => base64_encode($salt),
            'algorithm' => self::CIPHER
        ];
    }
```

Fungsi ini menerima dua parameter berupa string, yaitu plaintext sebagai data yang akan dienkripsi dan password sebagai kunci untuk proses enkripsi, serta mengembalikan sebuah array yang berisi data terenkripsi beserta parameter kriptografi yang dibutuhkan. Pertama, fungsi memanggil validateEncryptionInput untuk memastikan input valid. Kemudian dibuat initialization vector (IV) dan salt secara acak, di mana salt digunakan untuk menghasilkan kunci enkripsi melalui fungsi deriveKey dengan metode derivasi kunci tertentu. Setelah kunci dihasilkan, fungsi menggunakan openssl_encrypt untuk mengenkripsi data dengan algoritma yang didefinisikan di self::CIPHER dan IV yang telah dibuat. Jika proses enkripsi gagal, fungsi akan melempar exception RuntimeException dengan pesan kesalahan dari OpenSSL. Hasil akhirnya dikemas dalam bentuk array yang berisi ciphertext, IV, salt, dan algoritma, semuanya dalam bentuk base64 agar mudah disimpan atau dikirim. Fungsi ini menjamin keamanan data karena menggunakan IV dan salt acak serta kunci yang diturunkan dari password, sehingga setiap enkripsi untuk plaintext yang sama menghasilkan output yang berbeda.

Pada saat proses enkripsi, fungsi encrypt() menerima teks asli (*plaintext*) dan password. Fungsi ini membuat *salt* dan *IV (Initialization Vector)* acak, lalu menghasilkan *key* dari password dan *salt*. Data kemudian dienkripsi dengan

openssl_encrypt() menggunakan AES-128-CBC, dan hasilnya dikembalikan dalam bentuk array yang berisi *ciphertext*, *IV*, *salt*, dan informasi algoritma, semuanya dikodekan dalam base64 agar mudah dikirim atau disimpan.

4.1.3 Proses *Dekripsi Aes*-128

```
/**
 * Dekripsi lengkap data pengguna
 */
public function decryptPenggunaAll(array $userData): array
{
   $userData['alamat'] = $this->decryptAlamatPengguna($userData);
   $userData['telepon'] = $this->decryptTeleponPengguna($userData);
   return $userData;
}
```

Fungsi ini menerima parameter berupa array $userData, yang berisi informasi pengguna termasuk data terenkripsi seperti alamat dan nomor telepon. Di dalam fungsi, data alamat pengguna di*dekripsi* terlebih dahulu melalui pemanggilan metode decryptAlamatPengguna, kemudian nomor telepon di*dekripsi* melalui decryptTeleponPengguna. Hasil *dekripsi* tersebut kemudian disimpan kembali ke dalam array $userData pada key yang sesuai (alamat dan telepon). Akhirnya, fungsi mengembalikan array $userData yang sudah berisi data pengguna dalam bentuk plaintext, sehingga semua informasi sensitif yang terenkripsi dapat digunakan dengan aman di aplikasi. Fungsi ini memudahkan *dekripsi* beberapa field sekaligus dengan cara yang terstruktur dan terpusat.

```
/**
 * Dekripsi lengkap transaksi + pengguna
 */
public function decryptTransaksiWithPengguna(array $transaksiItem, array $userData): array
{
   // log_message('debug', 'DEBUG USER DATA: ' . print_r($userData, true));

   // Dekripsi alamat pengiriman jika tersedia
   if (!empty($userData['alamat_pengiriman']) && !empty($userData['iv_alamat']) && !empty($userData['salt_alamat'])) {
      $transaksiItem['alamat_pengiriman'] = $this->aesService->decrypt([
```

```
            'ciphertext' => $userData['alamat_pengiriman'],
            'iv' => $userData['iv_alamat'],
            'salt' => $userData['salt_alamat']
        ], $this->secretKey);
    }

    // Dekripsi data pengguna
    $userData = $this->decryptPenggunaAll($userData);

    // **Tambahkan alamat_pengiriman ke array yang dikembalikan**
    $userData['alamat_pengiriman'] = $transaksiItem['alamat_pengiriman'] ?? null;

    return array_merge($transaksiItem, $userData);
  }
}
```

Fungsi ini menerima dua parameter berupa array: $transaksiItem yang berisi informasi transaksi dan $userData yang berisi data pengguna yang mungkin terenkripsi.

Pertama, fungsi memeriksa apakah data alamat pengiriman tersedia serta memiliki IV dan salt yang terkait. Jika tersedia, alamat pengiriman di*dekripsi* menggunakan layanan AES (aesService->decrypt) dengan kunci rahasia ($this->secretKey) dan hasil *dekripsi* disimpan kembali di $transaksiItem['alamat_pengiriman']. Selanjutnya, semua data sensitif pengguna di*dekripsi* dengan memanggil metode decryptPenggunaAll, sehingga field seperti alamat dan nomor telepon berada dalam bentuk plaintext. Setelah itu, alamat pengiriman yang telah di*dekripsi* ditambahkan ke array $userData untuk memastikan semua informasi penting pengguna termasuk di hasil akhir.

Akhirnya, fungsi menggabungkan (array_merge) data transaksi dan data pengguna yang sudah di*dekripsi*, lalu mengembalikan array lengkap.

4.1.4 Proses *Watermark*

```
    /**
     * Apply watermark to target image
     */
    public function applyToImage(string $targetImagePath): bool
```

```
        {
            try {
                $watermark = $this->watermarkModel->getActiveWatermark();

                if (!$watermark || !$this->shouldApplyWatermark()) {
                    return false;
                }

                return $this->processWatermark($targetImagePath, $watermark);
            } catch (\Exception $e) {
                log_message('error', 'Watermark Error: ' . $e->getMessage());
                return false;
            }
        }
```

Fungsi ini menerima parameter berupa path gambar target ($targetImagePath) dan mengembalikan nilai boolean untuk menunjukkan keberhasilan proses. Proses dimulai dengan mencoba mengambil *watermark* aktif melalui model *watermarkModel*. Jika tidak ada *watermark* aktif atau kondisi untuk menerapkan *watermark* tidak terpenuhi (shouldApply*Watermark*() mengembalikan false), fungsi langsung mengembalikan false tanpa mengubah gambar. Jika *watermark* tersedia dan perlu diterapkan, fungsi memanggil *processWatermark* untuk menempelkan *watermark* pada gambar target. Seluruh proses dibungkus dalam blok try-catch untuk menangani potensi error; jika terjadi *exception*, pesan kesalahan dicatat menggunakan *log_message* dan fungsi mengembalikan false. Dengan demikian, fungsi ini memastikan *watermark* hanya diterapkan bila diperlukan dan aman terhadap kesalahan eksekusi.

```
/**
     * Main watermark processing logic
     */
    protected function processWatermark(string $targetImagePath, array $watermarkConfig): bool
    {
        $watermarkPath = $this->uploadPath . $watermarkConfig['path_gambar'];

        $this->validateFileExistence($targetImagePath, $watermarkPath);

        $mainImage = $this->createImageResource($targetImagePath);
        $wmImage = imagecreatefrompng($watermarkPath);
```

```
        $this->validateImageResources($mainImage, $wmImage);

        // Set alpha blending and save alpha for main image
        imagealphablending($mainImage, true);
        imagesavealpha($mainImage, true);

        // Set alpha blending and save alpha for watermark image
        imagealphablending($wmImage, true);
        imagesavealpha($wmImage, true);

        // Skip removeBlackWhiteBackground if watermark is already transparent
        if (!$this->hasTransparency($wmImage)) {
            $this->removeBlackWhiteBackground($wmImage);
        }

        $resizedWM = $this->resizeWatermark($wmImage, $watermarkConfig);
        $position = $this->calculatePosition($mainImage, $resizedWM,
$watermarkConfig);

        $this->mergeWatermark($mainImage, $resizedWM, $position,
$watermarkConfig);
        $this->saveImage($mainImage, $targetImagePath);

        $this->cleanupResources($mainImage, $wmImage, $resizedWM);

        return true;
    }
```

Proses dimulai dengan menentukan path lengkap *watermark* dari *konfigurasi* dan memvalidasi keberadaan *file* gambar target serta *watermark* menggunakan *validateFileExistence*. Gambar utama dibuka melalui createImageResource, sedangkan *watermark* dibuka sebagai PNG menggunakan imagecreatefrompng. Selanjutnya, fungsi memvalidasi bahwa kedua *resource* gambar berhasil dibuat (*validateImageResources*).

Untuk mendukung transparansi, fungsi mengaktifkan *alpha blending* dan menyimpan informasi *alpha* pada kedua gambar. Jika *watermark* belum transparan, fungsi akan menghapus *background* hitam-putih menggunakan *removeBlackWhiteBackground*. Setelah itu, *watermark* diubah ukurannya sesuai konfigurasi *(resizeWatermark*) dan posisi penempatannya dihitung relatif terhadap gambar utama (*calculatePosition*). *Watermark* kemudian digabungkan

ke gambar utama melalui *mergeWatermark,* dan hasil akhir disimpan ke path target (*saveImage*).

Terakhir, semua *resource* gambar dibersihkan dari memori menggunakan *cleanupResources* untuk mencegah kebocoran memori. Fungsi ini memastikan bahwa *watermark* diterapkan dengan benar, mempertahankan transparansi, ukuran, dan posisi yang sesuai, serta aman dari error. Hasil akhirnya adalah true jika semua langkah berhasil dilakukan.

4.1.5 Penggunaan *Aes* di dalam *controller*

```
public function decryptData()
{
    $request = $this->request;
    $id_transaksi = $request->getPost('id_transaksi');
    $field        = $request->getPost('field');
    $password     = $request->getPost('password');

    if (!$id_transaksi || !$field || !$password) {
        return $this->response->setJSON([
            'success' => false,
            'message' => 'Data tidak lengkap untuk dekripsi.'
        ]);
    }

    // Cek password
    $validPassword = getenv('ENCRYPTION_SECRET_KEY');
    if ($password !== $validPassword) {
        return $this->response->setJSON([
            'success' => false,
            'message' => 'Password salah.'
        ]);
    }

    try {
        $transaksi = $this->transaksiModel
            ->select('transaksi.*, pengguna.nama as nama_pengguna, pengguna.email as email_pengguna, pengguna.telepon as telepon_pengguna, pengguna.alamat as alamat_pengguna, pengguna.iv_alamatP, pengguna.salt_alamatP, pengguna.iv_telepon, pengguna.salt_telepon, transaksi.alamat_pengiriman, transaksi.iv_alamat, transaksi.salt_alamat')
            ->join('pengguna', 'pengguna.id_pengguna = transaksi.id_pengguna')
            ->where('transaksi.id_transaksi', $id_transaksi)
            ->first();
```

```
        if (!$transaksi) {
            return $this->response->setJSON([
                'success' => false,
                'message' => 'Transaksi tidak ditemukan.'
            ]);
        }

        // Lakukan dekripsi seperti sebelumnya
        $decrypted = '[Data tidak dapat didekripsi]';
        $userData = [
            'alamat' => $transaksi['alamat_pengguna'],
            'iv_alamatP' => $transaksi['iv_alamatP'],
            'salt_alamatP' => $transaksi['salt_alamatP'],
            'telepon' => $transaksi['telepon_pengguna'],
            'iv_telepon' => $transaksi['iv_telepon'],
            'salt_telepon' => $transaksi['salt_telepon'],
        ];

        switch ($field) {
            case 'alamat_pengiriman':
                $decrypted = $this->decryptService->decryptAlamatTransaksi($transaksi);
                break;
            case 'alamat_pengguna':
                $decrypted = $this->decryptService->decryptAlamatPengguna($userData);
                break;
            case 'telepon':
                $decrypted = $this->decryptService->decryptTeleponPengguna($userData);
                break;
            default:
                return $this->response->setJSON([
                    'success' => false,
                    'message' => 'Field dekripsi tidak valid.'
                ]);
        }

        return $this->response->setJSON([
            'success' => true,
            'decrypted_data' => $decrypted
        ]);
    } catch (\Exception $e) {
        return $this->response->setJSON([
            'success' => false,
            'message' => 'Gagal mendekripsi data: ' . $e->getMessage()
        ]);
    }
```

```
}
```

Fungsi decryptData() dalam sebuah *controller* (kemungkinan di *framework* seperti *CodeIgniter* atau *Laravel*) yang berfungsi untuk men*dekripsi* data sensitif transaksi dan pengguna, seperti *alamat pengiriman*, *alamat pengguna*, dan *nomor telepon pengguna*, menggunakan layanan keamanan berbasis AES (*Advanced Encryption Standard*).

Fungsi ini pertama-tama mengambil data dari request pengguna, yaitu id_transaksi, field, dan password. Jika salah satu dari parameter ini tidak diisi, sistem langsung mengembalikan respons JSON berisi pesan bahwa data tidak lengkap. Kemudian, sistem memeriksa validitas password dengan membandingkan input pengguna terhadap *secret key* yang disimpan di *environment* (*ENCRYPTION_SECRET_KEY*). Jika password tidak cocok, sistem menolak akses dengan pesan "Password salah."

Setelah password terverifikasi, sistem mengambil data transaksi dari *database*, termasuk informasi pengguna terkait seperti nama, email, alamat, dan telepon, beserta *IV (Initialization Vector)* dan *salt* yang digunakan saat enkripsi. Jika data transaksi tidak ditemukan, sistem mengembalikan pesan kesalahan.

Selanjutnya, sistem mempersiapkan array $userData yang berisi data terenkripsi milik pengguna. Berdasarkan nilai *field* yang dikirim dari *request* (misalnya alamat_pengiriman, alamat_pengguna, atau telepon), fungsi akan memanggil metode *dekripsi* yang sesuai dari *DecryptService*, seperti decryptAlamatTransaksi(), decryptAlamatPengguna(), atau decryptTeleponPengguna(). Jika *field* yang diminta tidak dikenal, sistem mengembalikan pesan bahwa *field dekripsi tidak valid*.

Akhirnya, jika proses *dekripsi* berhasil, sistem mengirimkan respons JSON berisi hasil *dekripsi* (*decrypted*_data). Namun, jika terjadi kesalahan selama proses, fungsi akan menangkap *exception* dan menampilkan pesan aman seperti “Gagal men*dekripsi* data”.

4.1.6 Penggunaan *Watermark* dalam *controller*

```
public function tambah(): ResponseInterface
    {
        if (!$this->validateTambahProduk()) {
                return $this->sendJsonResponse(false, implode(', ', $this->validator->getErrors()));
        }

        $gambar = $this->request->getFile('gambar');
        $namaGambar = null;

        try {
            $namaGambar = $this->handleFileUpload($gambar);
            $data = $this->prepareProductData($namaGambar);

            $this->applyWatermarkToImage($namaGambar);

            $this->produkModel->insert($data);
            $insertedId = $this->produkModel->getInsertID();

           return $this->sendJsonResponse(true, 'Produk berhasil ditambahkan', ['id' => $insertedId]);
        } catch (Exception $e) {
            $this->cleanUpUploadedFile($namaGambar);

              return $this->sendJsonResponse(false, 'Terjadi kesalahan sistem. Silakan coba lagi.');
        }
    }

    public function update(): ResponseInterface
    {
        $id = $this->request->getPost('id_produk');

        if (!$this->validateProductId($id)) {
            return $this->sendJsonResponse(false, 'ID produk tidak valid');
        }

        $produk = $this->getProdukById($id);
        if (!$produk) {
            return $this->sendJsonResponse(false, 'Produk tidak ditemukan');
        }
```

```
        // Validasi termasuk validasi gambar
        if (!$this->validateUpdateProduk()) {
                return $this->sendJsonResponse(false, implode(', ', $this->validator-
>getErrors()));
        }

        $gambarDiupload = false;
        $namaGambar = $produk['gambar'];
        $gambar = $this->request->getFile('gambar');

        // Handle file upload
        if ($gambar && $gambar->isValid() && !$gambar->hasMoved()) {
            $namaGambar = $this->handleFileUpload($gambar);
            $gambarDiupload = true;

            $this->applyWatermarkToImage($namaGambar);
        }

        try {
            $data = $this->prepareUpdateData($namaGambar, $gambarDiupload);
            $changes = array_diff_assoc($data, $produk);

            // PERUBAHAN PENTING: Anggap gambar saja sebagai perubahan yang
valid
            if (empty($changes) && !$gambarDiupload) {
                return $this->sendJsonResponse(false, 'Tidak ada data yang diubah');
            }

            // Tambahkan gambar jika ada upload
            if ($gambarDiupload) {
                $changes['gambar'] = $namaGambar;
                $this->deleteOldImage($produk['gambar']);
            }

            $result = $this->produkModel->update($id, $changes);

            if ($result) {
                $updatedData = $this->getProdukById($id);
                    return $this->sendJsonResponse(true, 'Produk berhasil diperbarui',
$updatedData);
            } else {
                // Rollback jika update gagal
                if ($gambarDiupload) {
                    $this->cleanUpUploadedFile($namaGambar);
                }
                return $this->sendJsonResponse(false, 'Gagal memperbarui produk');
            }
        } catch (Exception $e) {
            // Rollback jika exception
            if ($gambarDiupload) {
                $this->cleanUpUploadedFile($namaGambar);
            }

                return $this->sendJsonResponse(false, 'Terjadi kesalahan sistem: ' . $e-
```

```
    >getMessage());
        }
    }
```

Controller atau *service* untuk menambah dan memperbarui produk pada aplikasi penjualan baju adat berbasis *web*, dengan integrasi *watermark* pada gambar produk. Metode tambah() menangani penambahan produk baru, mulai dari validasi input, unggah gambar, menambahkan *watermark* melalui apply*Watermark*ToImage(), hingga menyimpan data ke *database*. Jika terjadi kesalahan, file yang diunggah dibersihkan dan respons JSON dikembalikan.

Metode update() menangani pembaruan produk, termasuk validasi ID produk, pengambilan data lama, unggah gambar baru, serta penerapan *watermark* jika gambar diperbarui. Kode ini juga memastikan hanya perubahan data yang valid yang disimpan, dan melakukan *rollback* atau membersihkan file baru jika terjadi kegagalan.

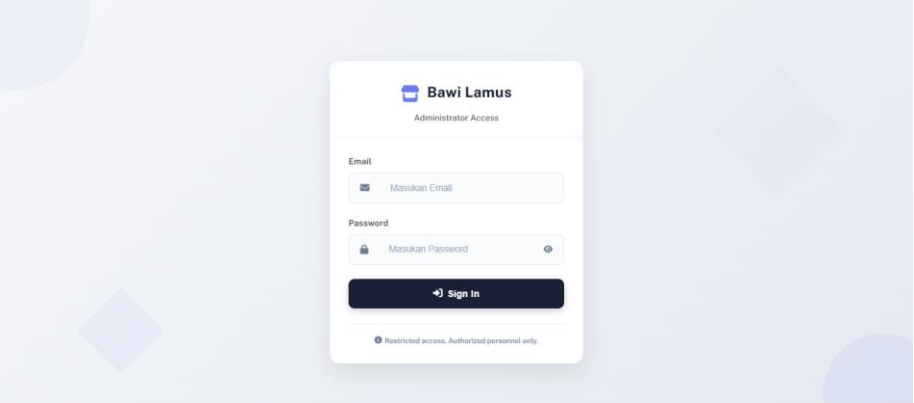
Fungsi apply*Watermark*ToImage() bertugas mengaplikasikan *watermark* ke gambar produk menggunakan *service WatermarkService*. Ia memeriksa keberadaan file, menerapkan *watermark* jika ada yang aktif, dan mencatat log aktivitas. Kesalahan pada proses *watermark* tidak menghentikan proses unggah produk sehingga integritas data tetap terjaga.

4.1.7 Fitur dan Halaman *Admin*

1. Login Admin

Proses *login* diawali dengan *validasi input username/email* dan *password*, lalu sistem melakukan pemeriksaan kredensial terhadap data yang tersimpan di *database*. *Password* biasanya disimpan dalam bentuk terenkripsi atau hash, sehingga saat *login* sistem membandingkan input *password* yang dimasukkan dengan hasil *dekripsi* atau hash dari *database*. Jika kredensial valid, sistem

memberikan *session* atau *token autentikasi* untuk *admin* agar dapat mengakses fitur-fitur manajemen, sementara jika gagal, sistem menampilkan pesan kesalahan. *Login* ini juga menjadi lapisan keamanan pertama sebelum *admin* dapat menambah produk, memperbarui data, menerapkan *watermark* pada gambar, atau melakukan *dekripsi* data transaksi dan pengguna, sehingga menjaga keamanan dan integritas sistem.

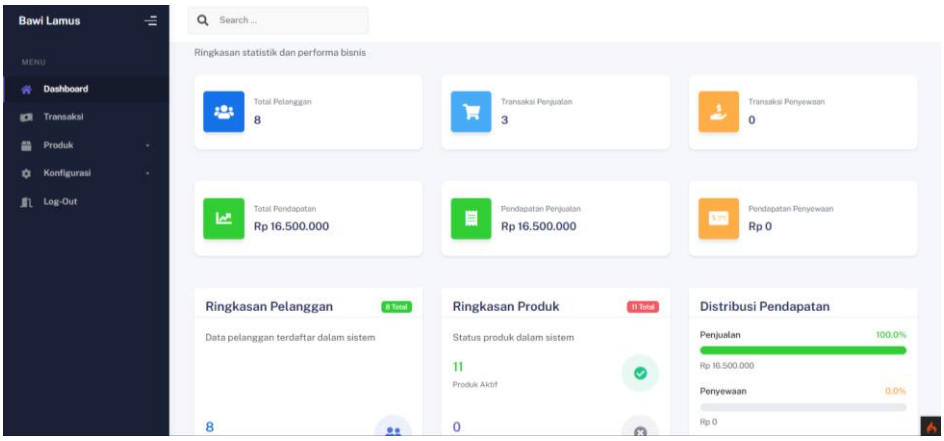


Gambar 4.1 *Login Admin*

2. Dashboard

Dashboard ini merupakan tampilan utama dari sistem Implementasi Keamanan Penjualan Baju Adat berbasis *Web* menggunakan *Watermarking* dan *Enkripsi AES*, yang menampilkan ringkasan statistik dan performa usaha secara keseluruhan. Di sebelah kiri terdapat menu navigasi untuk mengakses *Dashboard*, Transaksi, Produk, Konfigurasi, dan *Log-Out*, sedangkan area utama memperlihatkan informasi kunci seperti total pelanggan, jumlah transaksi penjualan, status penyewaan yang belum ada, total pendapatan, serta ringkasan produk. Bagian konfigurasi didesain untuk mengelola pengaturan sistem termasuk opsi keamanan seperti pengaturan *Watermarking* pada gambar produk dan pengelolaan kunci *Enkripsi AES* untuk melindungi data sensitif. Secara

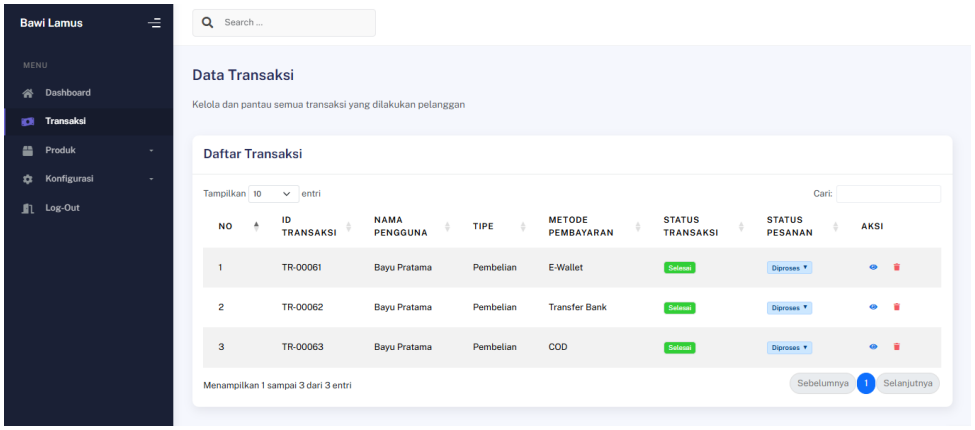
keseluruhan, *dashboard* ini tidak hanya memudahkan pemantauan aktivitas penjualan, pelanggan, dan inventaris, tetapi juga menjadi titik kontrol untuk memastikan aset digital dan data transaksi terlindungi melalui teknik *Watermarking* dan *Enkripsi AES*.



Gambar 4.2 *Dashboard*

3. Transaksi

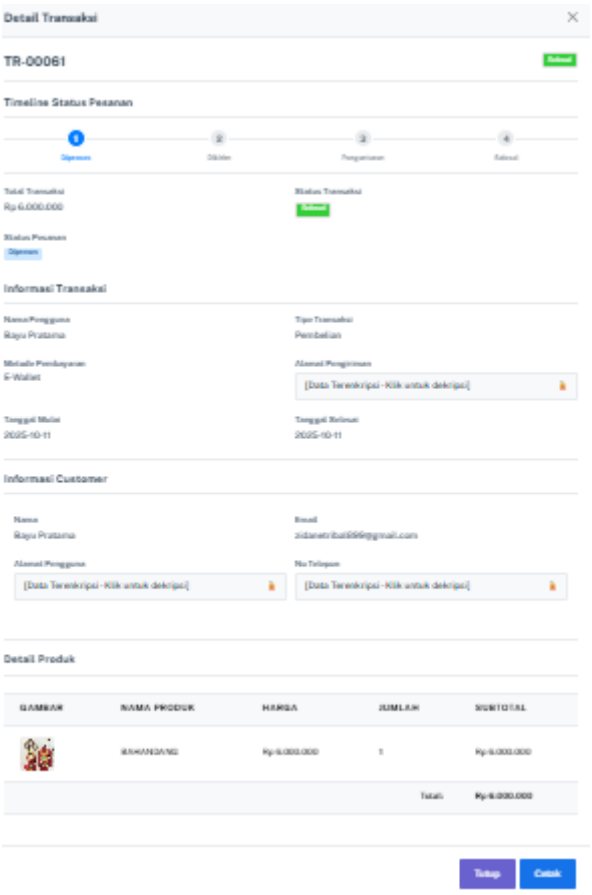
Halaman Data Transaksi dari sistem Implementasi Keamanan Penjualan Baju Adat Berbasis *Web* menggunakan *Watermarking* dan *Enkripsi AES*. Halaman ini berfungsi untuk mengelola serta memantau seluruh transaksi yang dilakukan oleh pelanggan. Pada bagian atas terdapat judul *Data Transaksi* disertai keterangan bahwa halaman ini digunakan untuk melihat dan memantau aktivitas pembelian. Di bawahnya terdapat tabel berisi daftar transaksi yang menampilkan beberapa kolom penting seperti Nomor Urut, ID Transaksi, Nama Pengguna, Tipe Transaksi, Metode Pembayaran, Status Transaksi, Status Pesanan, dan Aksi.



Gambar 4.3 Transaksi

4. Detail Transaksi

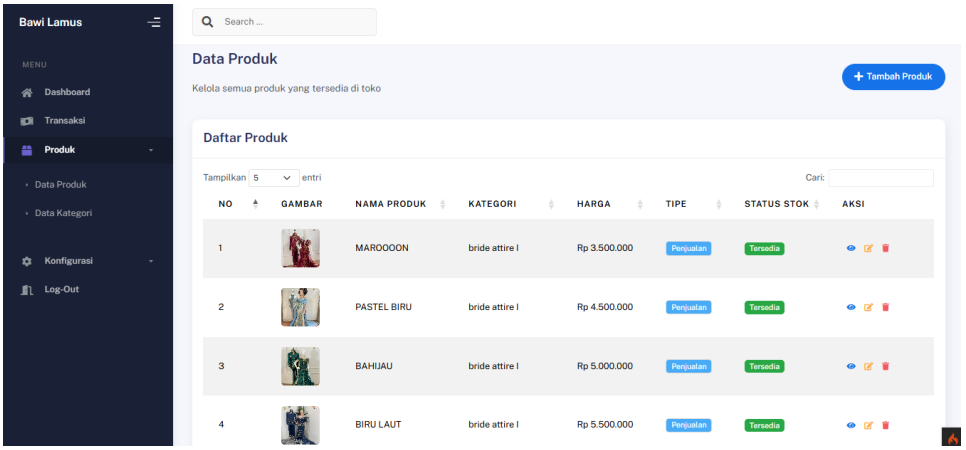
Halaman Detail Transaksi dari sistem Implementasi Keamanan Penjualan Baju Adat Berbasis *Web* menggunakan *Watermarking* dan *Enkripsi AES*. Halaman ini digunakan untuk menampilkan informasi lengkap mengenai transaksi. Pada bagian atas terdapat *timeline status pesanan* yang menunjukkan tahapan proses transaksi, mulai dari *Diproses*, *Dikirim*, *Pengiriman*, hingga *Selesai*, di mana posisi saat ini masih berada pada tahap *Diproses*.



Gambar 4.4 Detail Transaksi

5. Data Produk

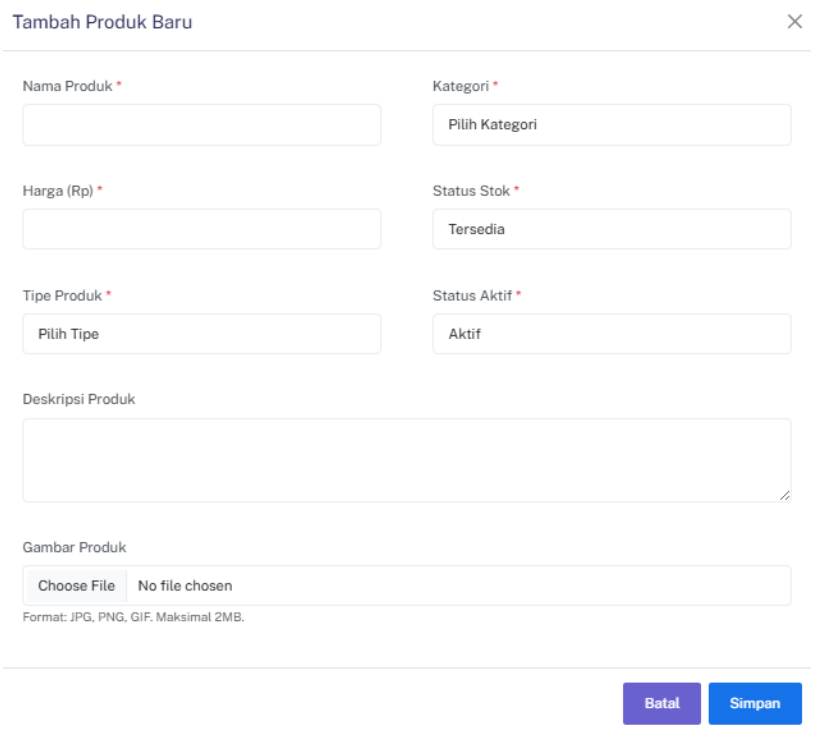
Halaman Data Produk dari sistem Implementasi Keamanan Penjualan Baju Adat Berbasis *Web* menggunakan *Watermarking* dan *Enkripsi AES*. Halaman ini berfungsi untuk mengelola seluruh produk yang tersedia di toko online baju adat. Pada bagian kiri terdapat menu navigasi utama seperti *Dashboard*, *Transaksi*, *Produk*, *Konfigurasi*, dan *Log-Out*. Sementara di bagian atas halaman terdapat judul Data Produk dengan keterangan bahwa halaman ini digunakan untuk mengelola semua produk yang dijual di toko, serta tombol Tambah Produk di pojok kanan atas yang memungkinkan admin menambahkan produk baru ke sistem.



Gambar 4.5 Data Produk

6. Form Tambah Produk

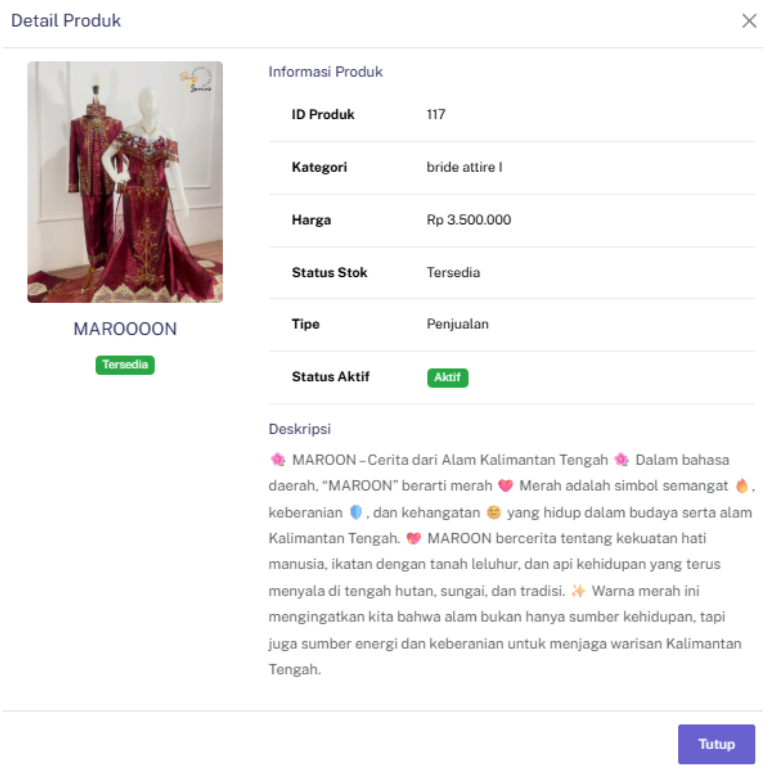
Tampilan halaman Tambah Produk Baru pada sistem Implementasi Keamanan Penjualan Baju Adat Berbasis *Web* menggunakan *Watermarking* dan *Enkripsi AES*. Halaman ini berfungsi untuk menambahkan data produk baru ke dalam database toko online baju adat. Dalam form tersebut terdapat beberapa kolom input yang wajib diisi oleh admin, antara lain Nama Produk, Kategori, Harga (Rp), Status Stok, Tipe Produk, Status Aktif, dan Deskripsi Produk. Selain itu, terdapat kolom untuk mengunggah gambar produk dengan ketentuan format file JPG, PNG, atau GIF, serta ukuran maksimal 2MB. Di bagian bawah, tersedia dua tombol, yaitu tombol Batal untuk membatalkan proses penambahan produk dan tombol Simpan untuk menyimpan data produk baru ke sistem.



Gambar 4.6 Form Tambah Produk

7. Form Detail Produk

Halaman ini menampilkan informasi lengkap tentang salah satu produk baju adat, seperti kategori baju. Di sisi kiri terdapat gambar produk dengan label status ketersediaan baju, sedangkan di sisi kanan ditampilkan detail informasi produk yang mencakup ID Produk, Kategori, Harga, Status Stok, Tipe, dan Status Aktif.

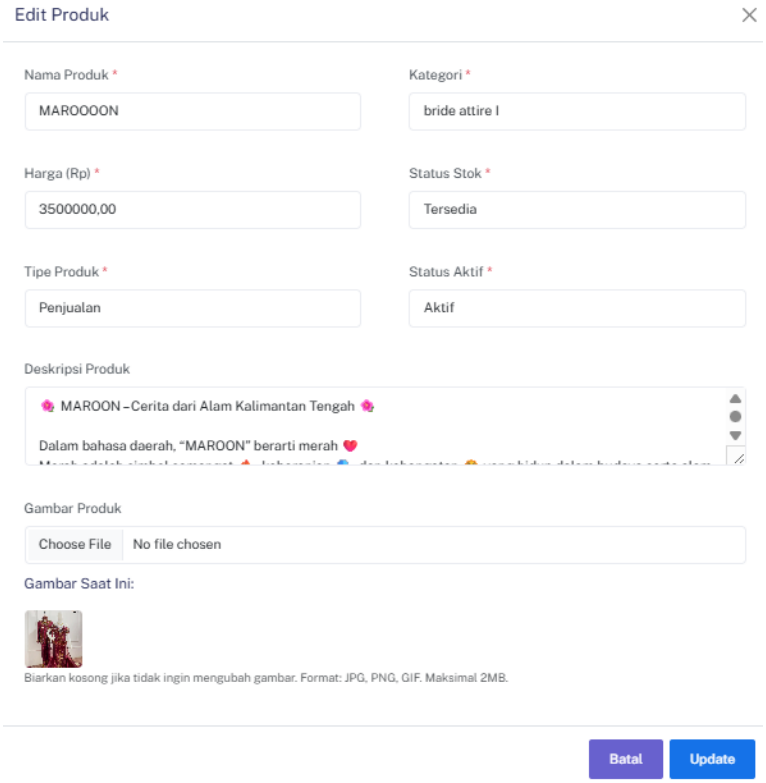


Gambar 4.7 Form Detail Produk

8. Form Edit Produk

Halaman ini digunakan oleh admin untuk memperbarui data produk yang sudah ada dalam sistem. Terdapat beberapa kolom yang dapat diedit, antara lain Nama Produk**,** Kategori**,** Harga (Rp)**,** Status Stok**,** Tipe Produk**,** Status Aktif, serta Deskripsi Produk**.**

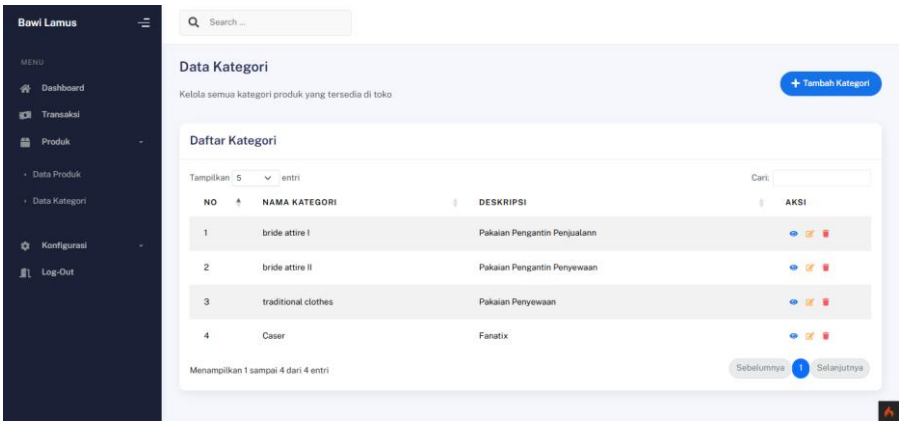
Selain kolom teks, halaman ini juga menyediakan fitur untuk mengganti gambar produk, di mana admin dapat mengunggah file gambar baru dengan format JPG, PNG, atau GIF berukuran maksimal 2MB. Namun, jika tidak ingin mengubah gambar, admin cukup membiarkan kolom tersebut kosong. Gambar produk saat ini juga ditampilkan di bagian bawah dengan label Gambar Saat Ini, agar admin dapat memastikan kesesuaian produk sebelum melakukan pembaruan.



Gambar 4.8 Form Edit Produk

9. Data Kategori

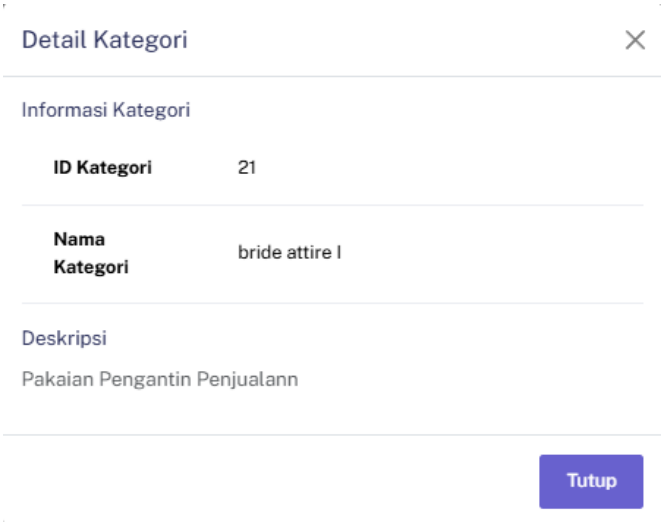
Halaman ini berfungsi untuk mengelola seluruh kategori produk yang tersedia di dalam toko daring. Tampilan dilengkapi dengan sidebar menu di sisi kiri yang berisi navigasi seperti Dashboard, Transaksi, Produk (Data Produk & Data Kategori), Konfigurasi, serta tombol Log-Out.



Gambar 4.9 Data Kategori

10. Form Detail Kategori

Halaman Detail Kategori pada sistem berfungsi untuk menampilkan informasi lengkap mengenai satu kategori produk yang tersimpan di dalam database. Dalam tampilan tersebut, admin dapat melihat data seperti ID Kategori, Nama Kategori, dan Deskripsi yang menjelaskan fungsi atau jenis produk yang termasuk dalam kategori tersebut, misalnya "bride attire I" dengan deskripsi "Pakaian Pengantin Penjualann." Halaman ini bersifat hanya untuk melihat data (*read-only*), ditandai dengan adanya tombol Tutup untuk kembali ke halaman sebelumnya tanpa melakukan perubahan.

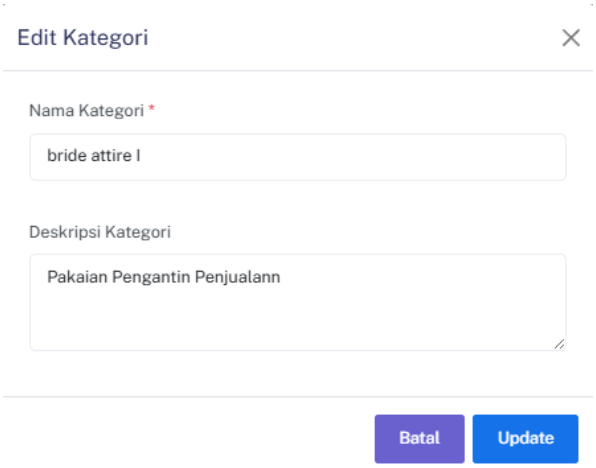


Gambar 4.10 Form Detail Kategori

11. Form Edit Kategori

Halaman Edit Kategori berfungsi untuk memperbarui data kategori produk yang telah tersimpan dalam sistem. Dalam tampilan ini, pengguna atau admin dapat mengubah informasi seperti Nama Kategori dan Deskripsi Kategori sesuai kebutuhan, misalnya memperbaiki kesalahan pengetikan atau memperbarui informasi agar lebih relevan. Setelah melakukan perubahan, admin dapat

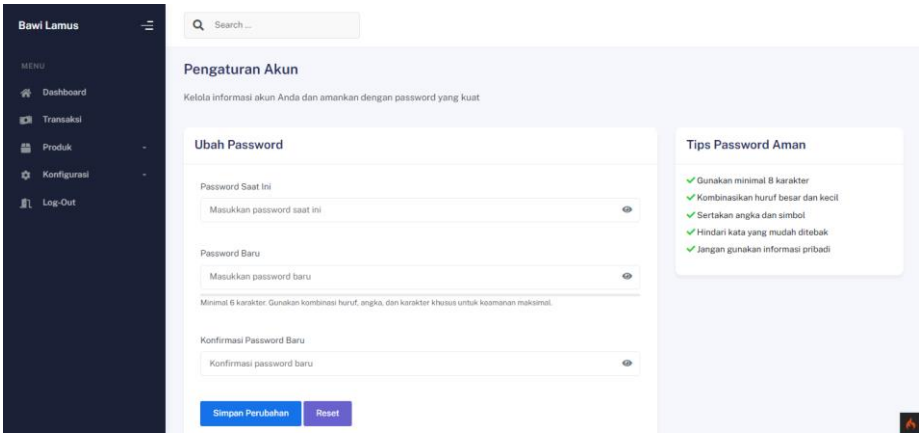
menekan tombol Update untuk menyimpan hasil editan, atau menekan Batal jika tidak ingin melanjutkan proses perubahan. Fitur ini mendukung kemudahan pengelolaan data pada sistem penjualan baju adat berbasis *web*, serta memastikan agar informasi kategori tetap akurat dan terkini.



Gambar 4.11 Form Edit Kategori

12. Konfigurasi Pengaturan Akun

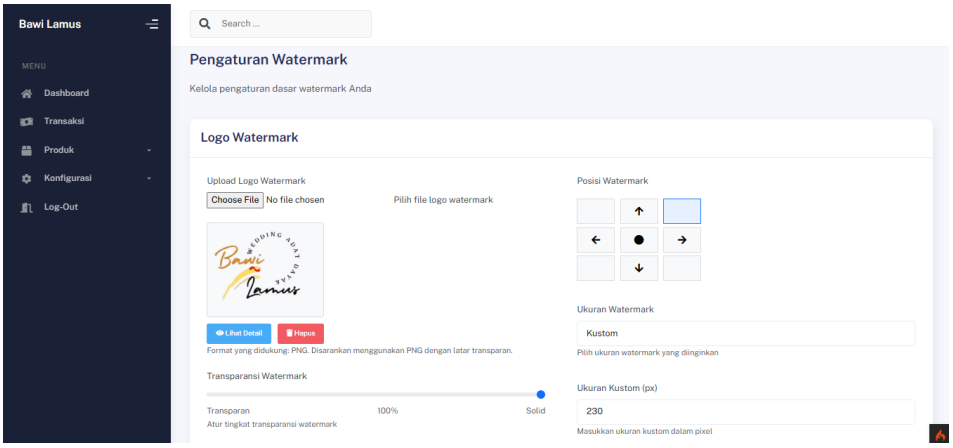
Halaman ini berfungsi untuk memberikan kendali penuh kepada pengguna terhadap keamanan akun mereka, khususnya melalui fitur ubah password. Pada bagian utama halaman, terdapat tiga kolom input: Password Saat Ini, Password Baru, dan Konfirmasi Password Baru. Pengguna diwajibkan memasukkan password lama sebagai langkah verifikasi identitas sebelum sistem mengizinkan perubahan password. Hal ini bertujuan untuk mencegah pihak lain yang tidak berwenang mengubah password tanpa sepengetahuan pemilik akun. Setelah password lama diverifikasi, pengguna dapat memasukkan password baru yang akan menggantikan yang lama, dan sistem akan memeriksa kecocokan antara kolom *Password Baru* dan *Konfirmasi Password Baru* untuk memastikan tidak ada kesalahan pengetikan.



Gambar 4.12 Konfigurasi Pengaturan Akun

13. Konfigurasi *Watermark*

Pengaturan *Watermark* pada sistem penjualan baju adat berbasis *web* berfungsi untuk mengatur tampilan dan karakteristik *watermark* yang akan diterapkan pada setiap gambar produk yang diunggah oleh penjual. Fitur ini merupakan bagian penting dari sistem keamanan visual yang bertujuan melindungi hak cipta dan keaslian gambar produk agar tidak disalahgunakan atau diklaim oleh pihak lain. Pada halaman ini, pengguna dapat mengunggah logo *watermark* dalam format PNG, yang direkomendasikan karena mendukung latar belakang transparan sehingga logo dapat menempel dengan baik di atas gambar tanpa menutupi detail produk. Setelah logo diunggah, pengguna dapat melihat pratinjau (melalui tombol *Lihat Detail*) atau menghapusnya jika ingin mengganti dengan logo baru (melalui tombol *Hapus*).

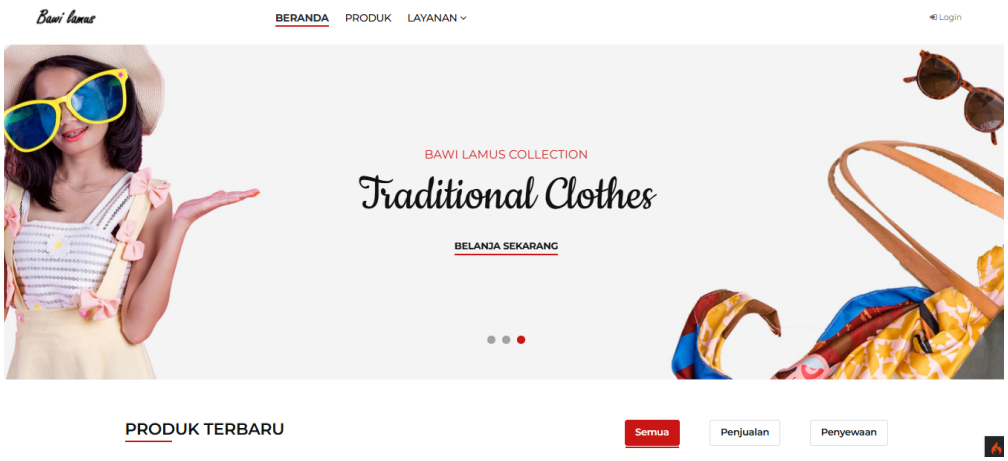


Gambar 4.13 Konfigurasi *Watermark*

4.1.8 Fitur Dan Halaman Pengguna

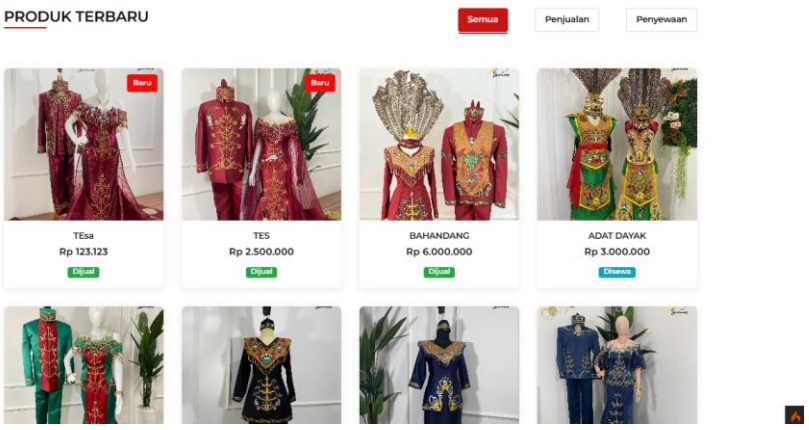
1. Halaman Utama Pelanggan

Halaman ini berfungsi sebagai tampilan utama yang menyambut pengunjung dan memperkenalkan identitas serta produk yang ditawarkan oleh platform. Desain halaman dibuat dengan tampilan modern dan bersih, menonjolkan kesan profesional serta mempermudah pengguna dalam menavigasi situs. Pada bagian atas (header), terdapat navigasi utama yang terdiri dari menu *Beranda*, *Produk*, *Layanan*, dan opsi *Login* di sisi kanan atas. Struktur ini membantu pengguna mengakses halaman lain seperti daftar produk, layanan penyewaan, atau masuk ke akun mereka dengan cepat.



Gambar 4.14 Halaman Utama Pelanggan

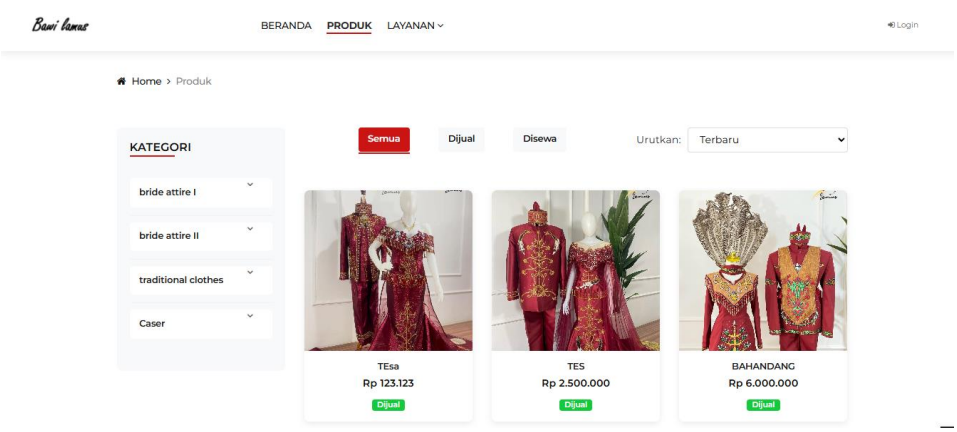
Menampilkan koleksi baju adat yang baru ditambahkan atau sedang dipromosikan kepada pengunjung. Setiap produk ditampilkan dalam bentuk kartu produk (product card) yang berisi foto pakaian, nama produk, harga, serta status ketersediaannya, apakah *Dijual* atau *Disewa*. Desain kartu ini dibuat sederhana dan rapi agar pengguna dapat dengan mudah membandingkan produk satu dengan lainnya. Selain itu, label berwarna seperti "Baru" dan "Dijual/Disewa" membantu pengguna mengenali informasi penting secara cepat tanpa harus membuka halaman detail produk.



Gambar 4.15 Halaman Produk Terbaru

2. Halaman Produk

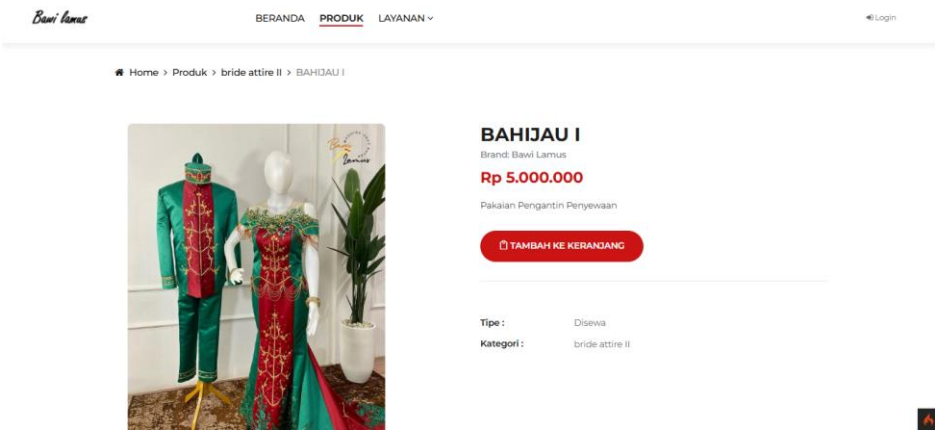
Halaman ini dirancang agar pengunjung dapat dengan mudah mencari, memfilter, dan membandingkan produk sesuai kategori dan kebutuhan mereka. Di bagian atas halaman terdapat navigasi utama dengan menu *Beranda*, *Produk*, dan *Layanan*, yang menunjukkan bahwa pengguna saat ini sedang berada di halaman *Produk*. Teks "Home > Produk" di bagian kiri atas berfungsi sebagai breadcrumb navigation, membantu pengguna mengetahui posisi halaman yang sedang mereka akses dalam struktur situs.



Gambar 4.16 Halaman Produk

3. Halaman Detail Produk

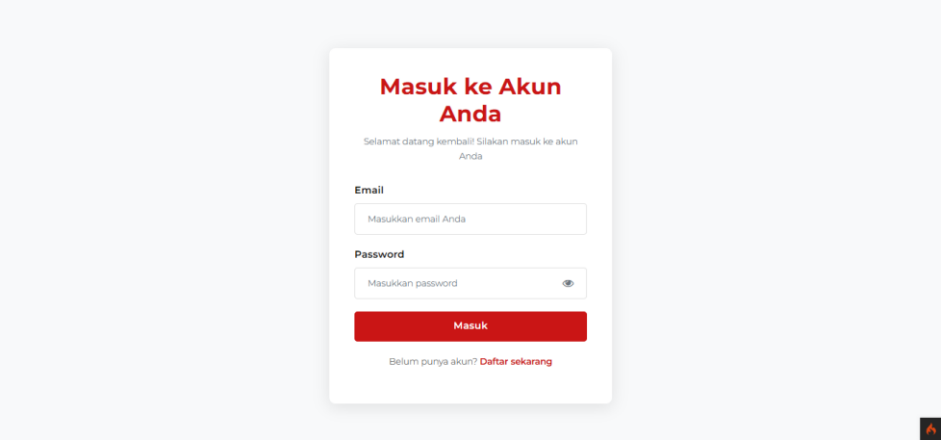
Halaman ini dirancang untuk memberikan gambaran menyeluruh tentang produk yang dipilih pengguna dari halaman daftar produk sebelumnya. Pada bagian atas, terdapat navigasi utama yang sama seperti halaman lain, *Beranda*, *Produk*, dan *Layanan*, dengan indikator aktif pada menu *Produk*, menandakan bahwa pengguna masih berada di dalam kategori produk. Tepat di bawahnya terdapat breadcrumb navigation: *Home* > *Produk* > *bride attire II* > *BAHIAU I*, yang membantu pengguna mengetahui lokasi hierarki produk di dalam sistem dan memudahkan untuk kembali ke kategori sebelumnya.



Gambar 4.17 Halaman Detail Produk

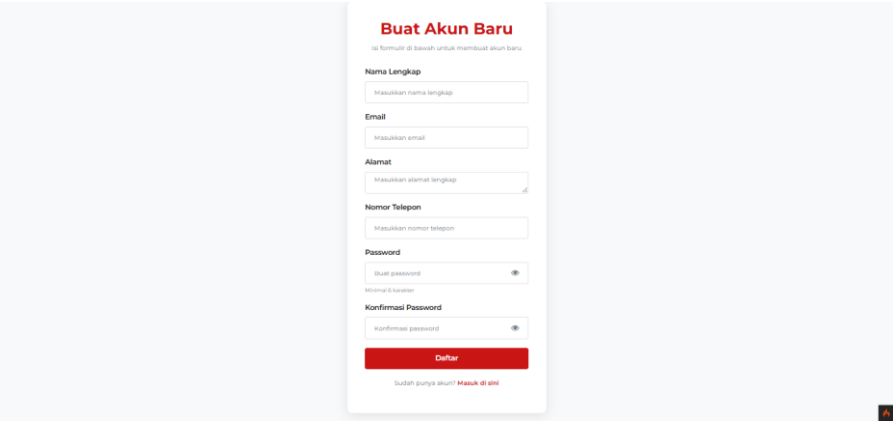
4. Halaman *Login* Dan *Register*

Halaman login atau masuk akun pengguna dalam sistem *web*. Tujuan utama halaman ini adalah untuk mengautentikasi pengguna sebelum mereka dapat mengakses fitur-fitur yang bersifat pribadi, seperti melihat riwayat transaksi, mengelola profil, atau melanjutkan proses pembelian dan penyewaan produk.



Gambar 4.18 Halaman *Login*

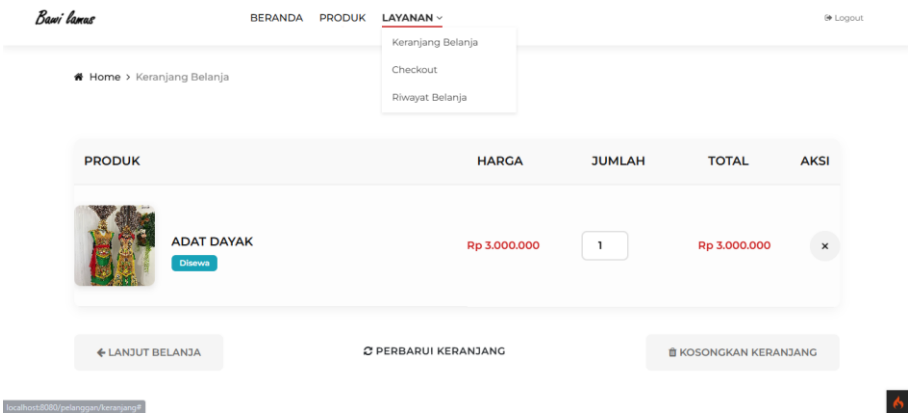
Halaman pendaftaran akun baru (register) pada sistem *web Bawi Lamus*. Halaman ini dirancang untuk memungkinkan pengguna baru membuat akun agar dapat menggunakan fitur-fitur yang membutuhkan autentikasi, seperti melakukan pembelian, penyewaan, atau pengelolaan profil.



Gambar 4.19 Halaman Register

5. Halaman Keranjang

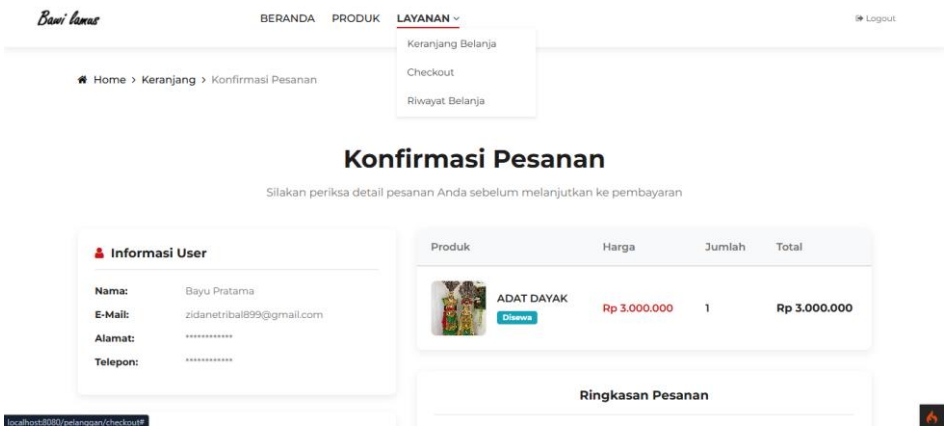
Menampilkan halaman “Keranjang Belanja” pada sistem *web Bawi Lamus*, yaitu bagian yang berfungsi untuk menampilkan daftar produk yang telah dipilih oleh pengguna sebelum melanjutkan ke proses pembayaran (*checkout*).



Gambar 4.20 Halaman Keranjang

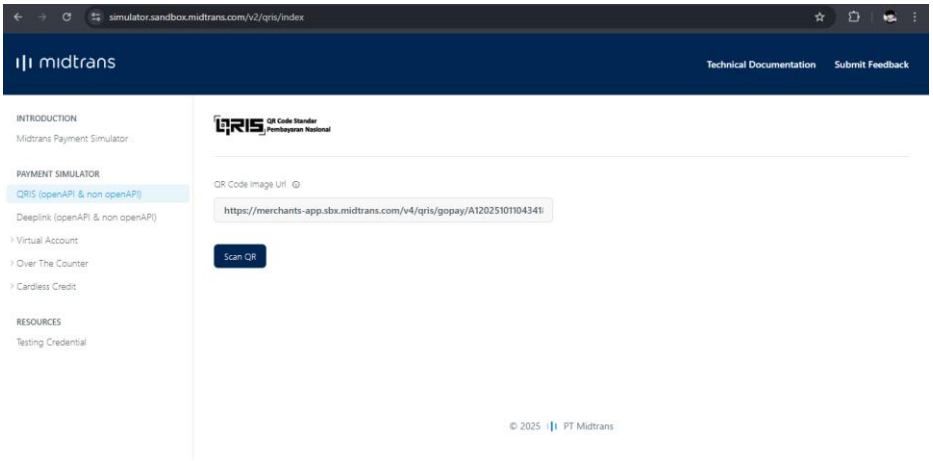
6. Halaman CheckOut

Halaman Konfirmasi Pesanan pada *web*site *Implementasi Keamanan Penjualan Baju Adat Berbasis Web Menggunakan Watermarking dan Enkripsi AES* berfungsi sebagai tahap akhir sebelum pengguna melanjutkan proses pembayaran. Pada halaman ini, pengguna dapat memeriksa kembali seluruh detail pesanan seperti nama produk, harga, jumlah, dan total biaya yang harus dibayar. Selain itu, ditampilkan juga informasi pengguna seperti nama, alamat email, alamat, dan nomor telepon, yang sebagian disembunyikan untuk menjaga kerahasiaan data pribadi. Tampilan halaman ini juga dilengkapi dengan menu navigasi di bagian atas yang berisi tautan ke halaman *Beranda*, *Produk*, serta *Layanan* yang mencakup *Keranjang Belanja*, *Checkout*, dan *Riwayat Belanja*.



Gambar 4.21 Konfirmasi Pesanan

Sistem pembayaran dalam proyek Implementasi Keamanan Penjualan Baju Adat Berbasis *Web* Menggunakan *Watermarking* dan *Enkripsi AES*, yang berfungsi untuk mensimulasikan pembayaran menggunakan metode QRIS (QR Code Standar Pembayaran Nasional). Pada tampilan tersebut, terlihat bahwa pengguna dapat memasukkan atau menggunakan URL gambar kode QR yang dihasilkan oleh Midtrans, kemudian menekan tombol “Scan QR” untuk menguji transaksi pembayaran melalui sistem QRIS atau GoPay.

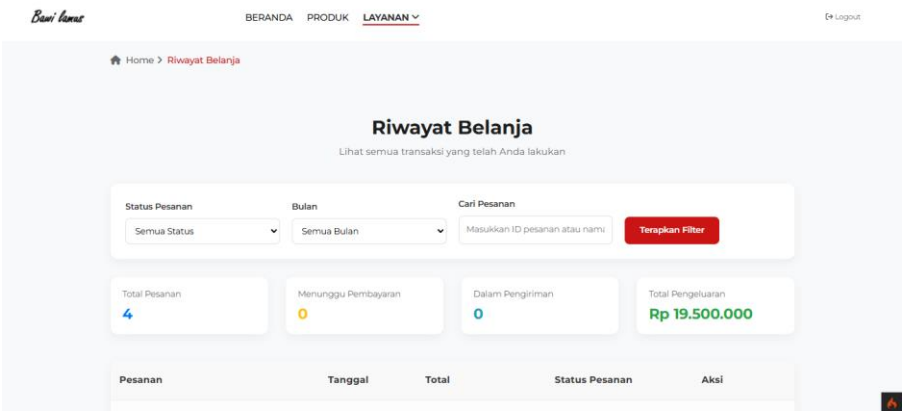


Gambar 4.22 Halaman Pembayaran

Halaman “Riwayat Belanja” dari *web*site *Implementasi Keamanan*

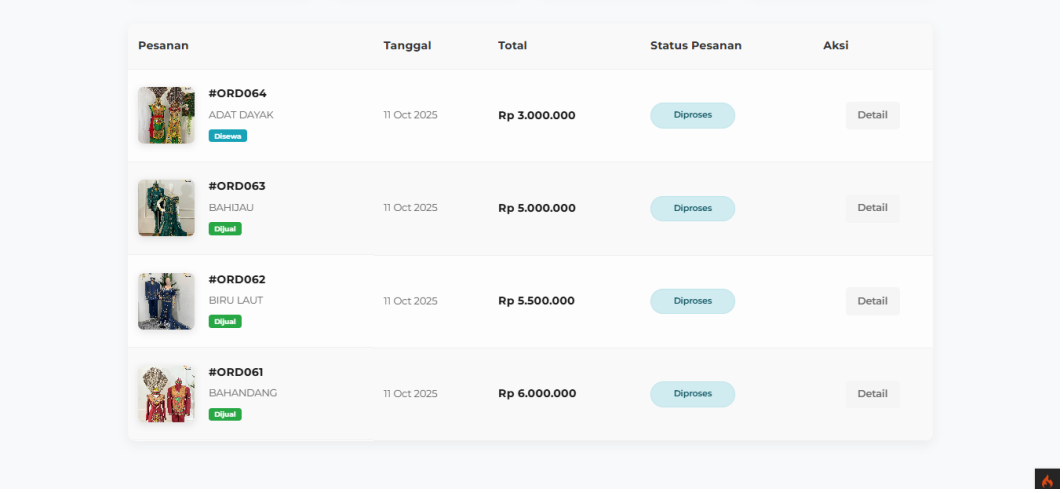
Penjualan Baju Adat Berbasis Web Menggunakan Watermarking dan Enkripsi AES.

Halaman ini berfungsi untuk menampilkan seluruh transaksi pembelian yang telah dilakukan oleh pengguna, lengkap dengan informasi status pesanan, tanggal transaksi, total biaya, serta opsi tindakan (aksi) seperti melihat detail pesanan.



Gambar 4.23 Halaman Riwayat Belanja

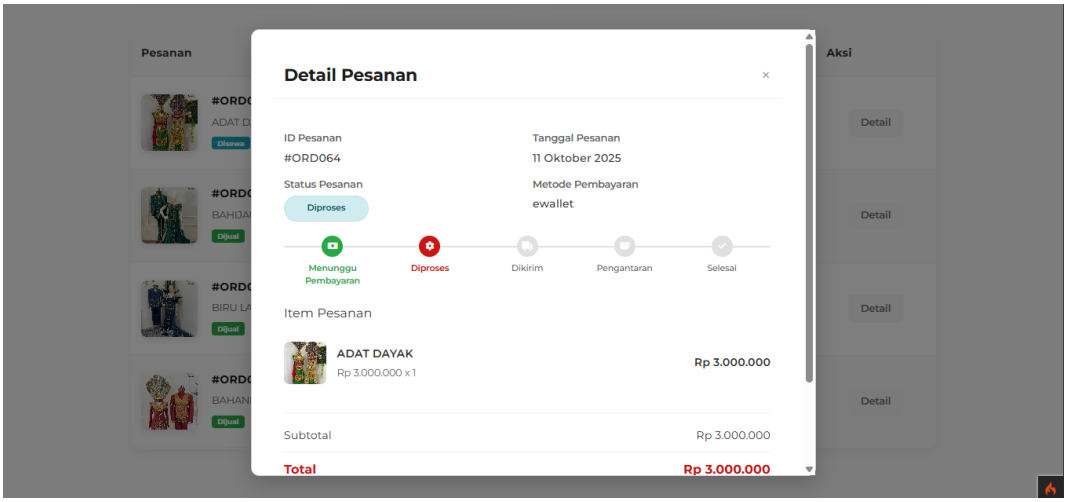
Sistem menampilkan daftar seluruh pesanan pengguna secara terstruktur dalam bentuk tabel yang berisi beberapa kolom penting, yaitu Pesanan, Tanggal, Total, Status Pesanan, dan Aksi. Setiap baris menunjukkan satu transaksi pembelian atau penyewaan baju adat.



Gambar 4.24 Daftar Pesanan

Halaman pop-up "Detail Pesanan" dari sistem *Implementasi Keamanan Penjualan Baju Adat Berbasis Web Menggunakan Watermarking dan Enkripsi AES*.

Halaman ini berfungsi untuk menampilkan informasi lengkap dari transaksi yang telah dilakukan oleh pengguna, dalam hal ini pesanan dengan ID #ORD064 untuk produk ADAT DAYAK. Di dalam tampilan ini, terdapat data penting seperti ID Pesanan, Tanggal Pesanan, Status Pesanan, serta Metode Pembayaran yang digunakan. Selain itu, terdapat alur status pesanan yang divisualisasikan melalui tahapan: *Menunggu Pembayaran* → *Diproses* → *Dikirim* → *Pengantaran* → *Selesai*, yang memudahkan pengguna memantau progres transaksi secara transparan.



Gambar 4.25 Detail Pesanan

4.1.9 Hasil Uji Coba

1. Uji Enkripsi/Deskripsi

Uji Enkripsi :

a. Pada saat daftar untuk menyimpan Nomor Telepon

```
8   INFO - 2025-10-17 10:14:37 --> [UJI-ENKRIPSI] Waktu: 408.682ms | Status: SUKSES
9   INFO - 2025-10-17 10:14:38 --> [UJI-ENKRIPSI] Waktu: 367.267ms | Status: SUKSES
```

Gambar 4.26 Uji *Enkripsi* Nomor Telepon

Pada saat daftar pengguna, sistem melakukan Uji Enkripsi Nomor Telepon menggunakan AES untuk memastikan keamanan data pribadi.

Proses tersebut berhasil dengan status *SUKSES* dan waktu eksekusi di bawah 0.5 detik, yang menunjukkan bahwa sistem mampu menyimpan data sensitif dengan aman dan efisien.

b. Pada saat *chekou*t menyimpan alamat pengguna dan alamat pengiriman

```
19  INFO - 2025-10-17 10:17:11 --> Metode pembayaran yang dipilih: cod
20  INFO - 2025-10-17 10:17:11 --> [UJI-ENKRIPSI] Waktu: 329.757ms | Status: SUKSES
```

Gambar 4.27 Uji *Enkripsi* Alamat *Chekout*

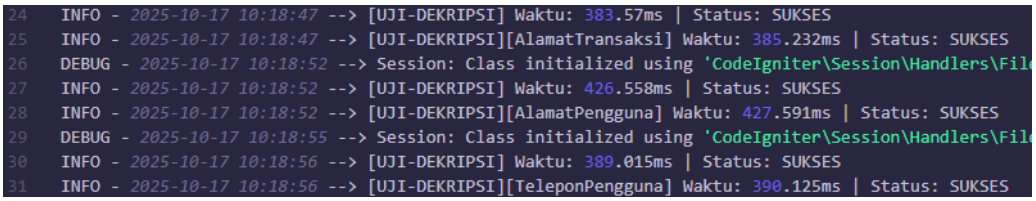
Pada tahap checkout, sistem melakukan Uji Enkripsi terhadap alamat pengguna dan alamat pengiriman sebelum disimpan ke database.

Hasil uji menunjukkan status SUKSES dengan waktu enkripsi sekitar 329 ms,

yang berarti sistem dapat memproses dan mengamankan data sensitif pengguna secara efisien dan berhasil.

Uji Deskripsi :

Pada saat deskrip Alamat pengguna, alamat pengiriman dan Nomor telepon



Gambar 4.28 Uji *Dekripsi*

Pada tahap *dekripsi*, sistem berhasil membuka kembali data yang telah terenkripsi, yaitu alamat pengguna, alamat pengiriman, dan nomor telepon dengan status sukses dan waktu eksekusi rata-rata di bawah 0,5 detik.

Hal ini menunjukkan bahwa mekanisme keamanan AES berfungsi dengan baik, baik untuk melindungi data saat disimpan maupun untuk mengembalikannya saat dibutuhkan.

2. Uji *Watermark*

Pada tahap pengujian *watermark*,dilakukan proses cropping pada gambar yang telah diberi *watermark* untuk melihat apakah *watermark* dapat dihilangkan tanpa merusak kualitas visual gambar. Hasil pengujian menunjukan bahwa ketika area gambar dipotong sebesar 10% pada bagian atas maupun kiri, *watermark* memang berkurang atau tidak terlihat, namun kualitas gambar secara keseluruhan menjadi tidak baik. Hal ini terlihat dari komposisi gambar yang tidak lagi utuh, objek terpotong, serta estetika visual yang menurun. Dengan kata lain, upaya menghapus *watermark* melalui cropping justru merusak tampilan gambar, sehingga *watermark* terbukti efektif dalam melindungi konten visual tanpa mudah dihilangkan oleh pengguna yang tidak berwenang .

Atas 10 %

Gambar Asli (Watermark)



Hasil Cropping 10% (top)



Kiri 10%

Gambar Asli (Watermark)



Hasil Cropping 10% (left)



Bawah 10%

Gambar Asli (Watermark)



Hasil Cropping 10% (bottom)



Kanan 10%

Gambar Asli (Watermark)



Hasil Cropping 10% (right)



Atas 20%

Gambar Asli (Watermark)



Hasil Cropping 20% (top)



Kiri 20%

Gambar Asli (Watermark)



Hasil Cropping 20% (left)



Bawah 20%

Gambar Asli (Watermark)



Hasil Cropping 20% (bottom)



Kanan 20%



Gambar 4.29 Percobaan Cropping

Tabel 4.1 Hasil Cropping

No	Posisi Cropping	Persentase	Keterangan
1	Atas	10%	Gambar dipotong dari bagian atas sebesar 10%
2	Kiri	10%	Gambar dipotong dari sisi kiri sebesar 10%
3	Bawah	10%	Gambar dipotong dari bagian bawah sebesar 10%
4	Kanan	10%	Gambar dipotong dari sisi kanan sebesar 10%
5	Atas	20%	Gambar dipotong dari bagian atas sebesar 20%
6	Kiri	20%	Gambar dipotong dari sisi kiri sebesar 20%
7	Bawah	20%	Gambar dipotong dari bagian bawah sebesar 20%
8	Kanan	20%	Gambar dipotong dari sisi kanan sebesar 20%

Sistem *Watermarking* yang diterapkan memiliki ketahanan baik terhadap serangan cropping hingga 20%, terutama jika posisi *watermark* dirancang strategis (misalnya di tengah atau tersebar diagonal).

Uji ini membuktikan bahwa mekanisme *watermark* mampu melindungi hak cipta gambar produk, bahkan ketika dilakukan modifikasi ringan seperti pemotongan sebagian.

4.1.10 Pengujian Sistem

Pada penelitian kali ini penulis melakukan pengujian menggunakan metode *blackbox*. *Blackbox* testing dilakukan dikampus STMIK Palangkaraya Pada tanggal 7 Oktober 2025 dengan penguji Ibu Amaya Andri Damaini, S.Kom., M.T. Pengujian *blackbox* dapat dilihat pada tabel berikut.

Tabel 4.2 Pengujian Blackbox Halaman Admin

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Menampilkan halaman login admin.	Sistem menampilkan halaman login admin.	Valid
Memasukan E-mail dan password jika salah.	Sistem menampilkan pesan login gagal jika E-mail atau password salah batas percobaan 5 kali.	Valid
Memasukan E-mail dan password jika benar.	Sistem langsung mengarahkan ke halaman dashboard admin.	Valid
Masuk kedalam halaman dashboard admin.	Sistem akan menampilkan halaman dashboard admin.	Valid
Memilih halaman transaksi.	Sistem akan menampilkan halaman transaksi.	Valid
Memilih detail transaksi	Sistem akan menampilkan data detail transaksi berdasarkan id yang dipilih.	Valid
Memilih status pesanan.	Sistem akan menampilkan option untuk status pesanan dan dapat berubah di halaman pengguna juga agar dapat melacak status pesanan.	Valid
Memilih deskripsi dan memasukan password untuk deskripsi jika salah.	Sistem akan menampilkan masukan password dan jika salah ada tampilan password salah.	Valid
Memilih deskripsi dan memasukan password untuk deskripsi jika benar.	Sistem akan menampilkan masukan password dan jika benar ada tampilan data berhasil dideskripsi.	Valid

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih hapus transaksi.	Sistem akan melakukan penghapusan berdasarkan id transaksi yang dipilih dan akan menampilkan pesan data transaksi berhasil dihapus.	Valid
Memilih halaman produk.	Sistem akan memunculkan dropdown yang ada data produk dan data kategori.	Valid
Memilih data produk	Sistem akan menampilkan halaman produk yang dapat melakukan CRUD pada data produk.	Valid
Melakukan penambahan data produk pada halaman data produk jika salah dan tidak sesuai dengan validasi.	Sistem akan memberikan form untuk mengisi data produk kemudian akan menampilkan pesan error pada bagian yang tidak sesuai dengan validasi.	Valid
Melakukan penambahan data produk pada halaman data produk jika benar.	Sistem akan memberikan form untuk mengisi data produk kemudian akan menampilkan pesan sukses kemudian data produk berhasil ditambahkan.	Valid
Memilih detail produk.	Sistem akan menampilkan data produk berdasarkan id produk yang dipilih.	Valid
Memilih update/pembaharuan pada data produk jika gagal memasukan data yang tidak sesuai dengan validasi.	Sistem akan memberikan form untuk melakukan update/pembaharuan jika gagal maka akan menampilkan pesan gagal karna tidak sesuai dengan validasi sistem.	Valid

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih menghapus produk berdasarkan id produk.	Sistem akan memberikan pilihan konfirmasi apakah produk akan dihapus, jika ya maka data produk akan terhapus dan jika batal maka tidak dihapus.	Valid
Memilih halaman data kategori.	Sistem akan menampilkan data kategori.	Valid
Melakukan tambah data pada kategori jika gagal.	Sistem akan menampilkan form untuk melakukan tambah data dan jika tidak sesuai dengan validasi sistem maka akan menampilkan pesan gagal menambahkan data kategori.	Valid
Melakukan tambah data pada kategori jika berhasil.	Sistem akan menampilkan form untuk melakukan tambah data dan jika berhasil dan sesuai validasi maka sistem akan menampilkan pesan sukses data kategori berhasil ditambahkan.	Valid
Memilih detail kategori berdasarkan id.	Sistem akan menampilkan detail kategori berdasarkan id yang dipilih.	Valid
Melakukan Update/Pembaharuan pada data kategori jika gagal.	Sistem akan menampilkan form untuk melakukan update/pembaharuan pada data kategori jika tidak sesuai dengan validasi sistem maka akan menampilkan pesan gagal menambahkan data.	Valid
Melakukan Update/Pembaharuan pada data kategori jika berhasil.	Sistem akan menampilkan form untuk melakukan update/pembaharuan pada data kategori jika sesuai dan validasi sistem benar maka akan menampilkan pesan data kategori berhasil diperbaharui.	Valid

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih halaman konfigurasi pada dashboard.	Sistem akan menampilkan dropdown atau pilihan menu antara pengaturan akun dan logo *watermark*.	Valid
Memilih halaman pengaturan akun	Sistem akan menampilkan form untuk melakukan perubahan password pada akun admin.	Valid
Memilih halaman pengaturan logo *watermark*.	Sistem akan menampilkan dan memberikan form *watermark*.	Valid
Melakukan penambahan logo *watermark* jika gagal.	Sistem akan menampilkan pesan gagal yang tidak sesuai dengan validasi sistem.	Valid
Melakukan penambahan logo *watermark* jika berhasil.	Sistem akan menampilkan pesan sukses jika sesuai dengan validasi sistem.	Valid
Memilih Logout	Sistem akan menampilkan pilihan untuk keluar, apakah ya atau batal. Jika ya maka akan mengarahkan ke halaman login admin kembali dan session berakhir jika batal maka tidak akan keluar dari halaman dashboard.	Valid

Tabel 4.2 Pengujian Blackbox Halaman Pengguna

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Menampilkan halaman home untuk bawi lamus.	Sistem akan menampilkan halaman dari penjualan bawi lamus.	Valid
Memilih halaman beranda.	Sistem akan menampilkan halaman awal dari bawi lamus dan data-data dari produk yang ada akan ditampilkan dihalaman ini.	Valid
Memilih filter penjualan atau penyewaan.	Sistem akan menampilkan data produk sesuai dengan filter yang dipilih.	Valid
Memilih lihat semua produk.	Sistem akan menampilkan halaman produk yang lebih banyak dan beragam serta dapat memilih berdasarkan kategori maupun berdasarkan tipe.	Valid
Memilih produk berdasarkan id produk.	Sistem akan menampilkan halaman untuk data detail produk berdasarkan id produk yang dipilih.	Valid
Memilih tambah ke keranjang.	Sistem akan menampilkan halaman keranjang dan dapat melakukan penambahan produk maupun langsung chekout.	Valid
Memilih lanjut belanja pada halaman keranjang.	Sistem akan menampilkan kembali halaman produk dan dapat melakukan penambahan kedalam keranjang.	Valid
Memilih hapus pada pilihan aksi di tabel keranjang	Sistem akan menampilkan pilihan untuk melakukan konfirmasi penghapusan data produk, jika ya maka data produk yang ada di keranjang akan terhapus dan jika batal data akan tetap ada.	Valid

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih lanjut ke chekout.	Sistem akan memeriksa apakah pelanggan sudah login atau belum, jika belum maka akan mengarahkan ke halaman login pelanggan dan data session akan disimpan.	Valid
Masuk ke halaman login pelanggan.	Sistem akan menampilkan halaman untuk melakukan login dan register akun jika tidak memiliki akun.	Valid
Melakukan pendaftaran akun pelanggan.	Sistem akan memberikan form untuk melakukan register pada halaman ini data pelanggan juga akan terenkripsi agar data tetap aman serta menjaga privasi pelanggan.	Valid
Melakukan pendaftaran akun pelanggan jika gagal.	Sistem akan menampilkan pesan gagal karna tidak sesuai dengan validasi sistem.	Valid
Melakukan pendaftaran akun pelanggan jika berhasil.	Sistem akan menampilkan pesan sukses dan mengarahkan ke halaman login pelanggan awal, akun anda berhasil dibuat. Data yang berhasil akan terenkripsi oleh AES-128.	Valid
Melakukan login pada halaman login pelanggan jika gagal.	Sistem akan menampilkan pesan gagal dan batas percobaan 5 kali.	Valid
Melakukan login pada halaman login pelanggan jika berhasil.	Sistem akan menampilkan halaman awal bawi lamus jika berhasil login, dan data session akan disimpan.	Valid
Melakukan chekout pada halaman keranjang.	Sistem akan menampilkan halaman chekout dan form untuk melengkapi proses penjualan maupun penyewaan.	Valid

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Melakukan pembayaran sesuai dengan yang dipilih.	Sistem akan menampilkan halaman payment gateway yang sesuai dengan tipe pembayaran yang dipilih oleh user.	Valid
Memilih halaman riwayat belanja.	Sistem akan menampilkan halaman dari riwayat belanja.	Valid
Memilih detail riwayat belanja.	Sistem akan menampilkan detail dari riwayat belanja.	Valid
Memilih logout.	Sistem akan menghapus session dan mengarahkan ke halaman login pelanggan lagi dan menampilkan pesan sukses.	Valid

4.2 Pembahasan

4.2.1. Hasil Pengujian

Berdasarkan hasil pengujian sistem, semua skenario menunjukkan hasil sesuai dengan yang diharapkan. Sistem berhasil menampilkan halaman login admin, membatasi percobaan login hingga lima kali, dan mengarahkan pengguna yang sah ke dashboard, sehingga efektif mencegah akses tidak sah. Kontrol akses data juga berjalan baik, di mana detail transaksi, produk, dan kategori hanya dapat diakses setelah login, serta operasi tambah, update, dan hapus data hanya berhasil jika input valid sesuai dengan mekanisme validasi sistem. Data sensitif dilindungi dengan mekanisme *dekripsi* yang memerlukan password, sedangkan penghapusan data selalu disertai konfirmasi. Selain itu, pengaturan akun dan *watermark* hanya dapat diakses oleh admin, dan proses logout mengakhiri sesi sehingga mencegah akses berkelanjutan. Dengan demikian, sistem terbukti efektif menahan serangan dasar dan mampu melindungi data pengguna serta menjaga integritas dan keamanan informasi secara keseluruhan.

4.2.2. Evaluasi Sistem

Evaluasi sistem dilakukan untuk menilai sejauh mana *web*site *Implementasi Keamanan Penjualan Baju Adat Berbasis Web Menggunakan Watermarking dan Enkripsi AES* berjalan sesuai dengan tujuan perancangannya, yaitu memberikan keamanan data transaksi dan perlindungan hak cipta digital pada media gambar produk. Berdasarkan hasil pengujian, sistem mampu menjalankan seluruh fungsinya dengan baik, meliputi proses pendaftaran pengguna, pemesanan produk, konfirmasi pesanan, riwayat transaksi, serta pembayaran melalui integrasi Midtrans (QRIS dan e-wallet). Proses *Enkripsi AES* berhasil melindungi data sensitif seperti alamat, nomor telepon, dan detail transaksi agar tidak dapat dibaca oleh pihak yang tidak berwenang. Selain itu, penerapan *Watermarking* pada gambar produk adat terbukti efektif dalam menjaga orisinalitas dan mencegah pencurian gambar di platform digital. Dari sisi antarmuka, sistem memiliki tampilan yang sederhana, responsif, dan mudah dipahami oleh pengguna. Dengan demikian, sistem dinilai layak untuk digunakan pada tahap implementasi lebih lanjut di lingkungan produksi, meskipun masih memerlukan beberapa penyempurnaan untuk meningkatkan efisiensi dan keamanannya secara menyeluruh.

4.2.3. Kelebihan Dan Kekurangan Sistem

1. Kelebihan Sistem
    a. Keamanan Data Terjamin**:** Penggunaan algoritma AES (*Advanced Encryption Standard)* berhasil melindungi data pengguna dan transaksi agar tidak mudah diakses oleh pihak yang tidak berwenang.

b. Perlindungan Gambar Produk: Implementasi digital *Watermarking* mencegah pencurian atau penyalahgunaan foto baju adat yang menjadi identitas budaya daerah.

c. Integrasi Pembayaran Aman: Sistem terhubung dengan *gateway* pembayaran Midtrans menggunakan QRIS dan e-wallet, sehingga proses pembayaran berjalan cepat, aman, dan terverifikasi.

d. Antarmuka Responsif dan User *Friendly*: Desain *web*site mudah digunakan baik melalui komputer maupun perangkat seluler, memudahkan pengguna dalam melakukan transaksi.

e. Pengelolaan Transaksi Lengkap: Fitur seperti konfirmasi pesanan, riwayat belanja, dan detail pesanan memberikan transparansi penuh terhadap aktivitas pembelian pengguna.

2. Kelemahan Sistem

a. Belum Sepenuhnya Otomatis: Beberapa proses seperti validasi pembayaran atau pembaruan status pengiriman masih memerlukan intervensi admin.

b. Ketergantungan pada Koneksi Internet: Sistem berbasis *web* membutuhkan koneksi yang stabil, sehingga pengguna dengan jaringan lambat akan mengalami kendala akses.

c. Belum ada Enkripsi End-to-End untuk File Media: Meskipun data teks sudah di*Enkripsi AES*, file gambar dengan *watermark* belum dienkripsi secara menyeluruh.

d. Pengujian terbatas dilLingkungan *Sandbox*: Proses pembayaran masih diuji pada mode *sandbox* Midtrans, belum diuji penuh pada mode produks

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa :

1. Sistem Penjualan Baju Adat Berbasis *Web* berhasil dibangun dan berjalan sesuai dengan tujuan penelitian, yaitu menyediakan platform penjualan baju adat yang aman dan informatif.
2. Penerapan algoritma *Enkripsi AES* (*Advanced Encryption Standard*) terbukti efektif dalam menjaga kerahasiaan data sensitif pengguna seperti alamat, nomor telepon, dan detail transaksi agar tidak mudah diakses oleh pihak tidak berwenang.
3. Penggunaan teknologi digital *Watermarking* mampu melindungi gambar produk baju adat dari tindakan plagiarisme atau pencurian visual, sehingga hak cipta digital dapat terjaga.
4. Integrasi sistem pembayaran melalui Midtrans (QRIS dan e-wallet) berhasil diuji dengan baik dalam lingkungan sandbox, menunjukkan bahwa transaksi dapat berlangsung secara aman, cepat, dan terverifikasi.
5. Sistem menyediakan fitur utama seperti konfirmasi pesanan, riwayat belanja, dan detail transaksi, yang memudahkan pengguna dalam memantau aktivitas pembelian.
6. Antarmuka *web*site dirancang dengan tampilan user friendly dan responsif, sehingga mudah digunakan pada berbagai perangkat.
7. Berdasarkan hasil pengujian, sistem ini dinilai layak untuk diterapkan

8. sebagai solusi digital dalam mendukung promosi dan penjualan baju adat secara aman serta modern.

5.2 Saran

Agar sistem dapat dikembangkan dan diimplementasikan lebih optimal di masa mendatang, beberapa saran yang dapat diberikan antara lain:

1. Integrasi pada Lingkungan Produksi: Sistem pembayaran perlu diuji dan diaktifkan pada mode produksi Midtrans agar dapat digunakan untuk transaksi nyata.
2. Peningkatan Keamanan File Media: Dapat ditambahkan mekanisme enkripsi file gambar dan dokumen transaksi agar keamanan data lebih menyeluruh, tidak hanya pada teks.
3. Optimasi Performa Sistem: Penggunaan caching dan kompresi file dapat meningkatkan kecepatan akses halaman, terutama untuk pengguna dengan koneksi internet terbatas.
4. Pengembangan Aplikasi Mobile: Untuk memperluas jangkauan pengguna, sistem ini dapat dikembangkan dalam bentuk aplikasi Android/iOS yang terhubung langsung dengan server utama.
5. Sistem diharapkan dapat dikembangkan dengan menambahkan mekanisme enkripsi pada data sensitif seperti nama dan ID pembeli
6. Disarankan untuk menggunakan jenis *watermark digital* yang mampu menyisipkan informasi tersembunyi (*embedded digital data*) pada gambar produk

DAFTAR PUSTAKA

Aryanto, W., & Kasnawi. (2025, Februari). Penerapan *Advanced Encryption Standard* Pada Aplikasi *Payment Gateway* Untuk Keamanan Data Transaksi Penjualan.

Defta, P., Y., & Agus, P. (2022, April). *Implementasi Digital Watermarking* Pada *File Audio* Mp3 Menggunakan Metode FHSS (*Frequency Hopping Spread Spectrum*) Dan Fourier Transform.

Eka, A., Felica, A., Jonathan, S., Zikri, K., M., N., Razan, & Ratnawati. (2023, Juni). Perancangan *Website* Penjualan Pupuk dan Bibit Bersubsidi pada Dinas Pertanian Karawang.

Fauzan, M. R., & Ariyani, P. (2024, September). Penerapan Kriptografi *AES*-128 Untuk Keamanan Data Berbasis *Website* Pada Cahaya Battery. In Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (*SENAF TI*) (Vol. 3, No. 2, pp. 163–171).

Hardianti, W., Joni, K., & Elmayati. (2023, Juni). *Implementasi Algoritma AES* (*Advanced Encryption Standard*) Pada Penjualan Alat-Alat Berbasis *Web* (Studi Kasus Toko Sonia Elektronik).

Marsya C. K., Haifaturrahmah, & Inang, I. R. (Desember, 2025). Melestarikan Warisan Budaya Melalui Baju Adat di Sekolah Dasar.

M. Yusuf, Sutrisno, PA Andien P., M. Asir, & P. A. Cakranegara. (2022, Desember). Prospek Penggunaan *E-Commerce* Terhadap Profitabilitas dan Kemudahan Pelayanan Konsumen: *Literature Review*.

Mufti, M., Wiharto, Y., & Subandi, S. (2025). Penerapan Algoritma Kriptografi *Advanced Encryption Standard* 128 Pada Laporan Transaksi di Kopi Tuku. *Journal Software, Hardware and Information Technology*, 5(1), 48–60.

Narulita, Siska, A. Nugroho, & M. Z. Abdillah. (2024). *Diagram Unified Modelling Language (UML)* untuk Perancangan *Sistem Informasi* Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, 2(3), 244–256.

Nilfaidah., & Najmi. (2021). Pengembangan Sistem Absensi Mahasiswa *Realtime* Menggunakan *PHP*, *MySQL*, SMS Gateway, dan *Framework CodeIgniter*. (Skripsi). Universitas Negeri Makassar.

Nugroho, R. A., Suarna, N., Ali, I., & Efendi, D. I. (2025). Penerapan Algoritma FP-Growth Untuk Optimalisasi Pola Asosiasi Dalam Data Transaksi Penjualan Obat: Studi Kasus di Apotek XYZ. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(1).

Pita S., & Jauhar A. (2021). *Implementasi Program* Keluarga Harapan Dilihat dari Aspek Komunikasi di Desa Mahe Seberang Kecamatan Haruai.

Putri, M. P., Nadeak, E., Malahayati, M., Rahmi, N., Rini, A., Sari, D. N., ... & Pratama, R. A. A. (2023). *Sistem Manajemen Basis Data* Menggunakan *MySQL*.

R. Oktaviandi, H. Zakaria, A. U. Zailani, R. Ilyasah, D. S. Kamil, Lindawati, M Asrizal, M. F. Haidar, P. R. Salim, S. Falah, & Y. Gibran. (2024, April). Pengenalan Konsep Perancangan Perangkat Lunak Menggunakan *Draw.io* Bagi Siswa SMK Sebagai Persiapan Karir Menjadi Programmer.

Rizky, M. R., & Fatimah, T. (2023, Oktober). Implementasi Kriptografi Algoritme *AES*-128 Untuk Pengamanan Data Penjualan dan Pembelian Mobil Pada Showroom Bob's Auto. In Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (*SENAF TI*) (Vol. 2, No. 2).

Sidabutar, L., Ramadhan, M., & Panjaitan, Z. (2024). Implementasi Kriptografi Pengamanan Data Pemesanan Produk Menggunakan Metode *AES*. *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, 3(4), 440–449.

S. Asri, T. Peryanto, E. Malays, Tif., & U. Yai. (Oktober 2024). Perbandingan Impelmentasi Algoritma *AES* Dalam Pemograman Jaringan dan Analisis Algoritma Enkripsi Untuk Pengamanan Komunikasi Jaringan.

Waruwu, M. (2024). Metode Penelitian dan Pengembangan (*R&D*): Konsep, Jenis, Tahapan dan Kelebihan. *Jurnal Ilmiah Profesi Pendidikan*, 9(2), 1220–1230.

Winata, A. (2025). Penerapan *Advanced Encryption Standard* pada Aplikasi *Payment Gateway* Untuk Keamanan Data Transaksi Penjualan. *Techno.com*, 24(1).

LAMPIRAN

Proses *Enkripsi AES*-128

```
<?php

namespace App\Services;

use RuntimeException;
use InvalidArgumentException;

class AesPasswordService
{
    private const CIPHER = 'aes-128-cbc';
    private const KEY_LENGTH = 16; // 128-bit
    private const ITERATIONS = 100000;
    private const SALT_LENGTH = 16;
    private const MIN_PASSWORD_LENGTH = 1;

    private int $ivLength;

    public function __construct()
    {
        $this->validateEnvironment();
        $this->ivLength = openssl_cipher_iv_length(self::CIPHER) ?: 0;
    }

    /**
     * Enkripsi data dengan password
     */
    public function encrypt(string $plaintext, string $password): array
    {
        $this->validateEncryptionInput($plaintext, $password);

        $iv = $this->generateRandomBytes($this->ivLength);
        $salt = $this->generateRandomBytes(self::SALT_LENGTH);
        $key = $this->deriveKey($password, $salt);

        $ciphertext = openssl_encrypt(
            $plaintext,
            self::CIPHER,
            $key,
            OPENSSL_RAW_DATA,
            $iv
        );

        if ($ciphertext === false) {
            throw new RuntimeException('Encryption failed: ' .
openssl_error_string());
        }

        return [
            'ciphertext' => base64_encode($ciphertext),
```

```
            'iv' => base64_encode($iv),
            'salt' => base64_encode($salt),
            'algorithm' => self::CIPHER
        ];
    }

    /**
     * Dekripsi data terenkripsi
     */
    public function decrypt(array $encryptedData, string $password): string
    {
        $this->validateDecryptionInput($encryptedData, $password);

        $iv = $this->decodeBase64($encryptedData['iv'], 'IV');
        $salt = $this->decodeBase64($encryptedData['salt'], 'salt');
        $ciphertext = $this->decodeBase64($encryptedData['ciphertext'], 'ciphertext');

        $this->validateDataLengths($iv, $salt, $ciphertext);

        $key = $this->deriveKey($password, $salt);

        $plaintext = openssl_decrypt(
            $ciphertext,
            self::CIPHER,
            $key,
            OPENSSL_RAW_DATA,
            $iv
        );

        if ($plaintext === false) {
            throw new RuntimeException('Decryption failed. Wrong password or corrupted data.');
        }

        return $plaintext;
    }


    /**
     * Generate key dari password dengan PBKDF2
     */
    private function deriveKey(string $password, string $salt): string
    {
        $key = hash_pbkdf2(
            'sha256',
            $password,
            $salt,
            self::ITERATIONS,
            self::KEY_LENGTH,
```

```
            true
        );

        if ($key === false) {
            throw new RuntimeException('Key derivation failed');
        }

        return $key;
    }

    /**
     * Validasi environment
     */
    private function validateEnvironment(): void
    {
        if (!extension_loaded('openssl')) {
            throw new RuntimeException('OpenSSL extension is not available');
        }

        if (!in_array(self::CIPHER, openssl_get_cipher_methods())) {
            throw new RuntimeException('Cipher ' . self::CIPHER . ' is not available');
        }
    }

    /**
     * Validasi input untuk enkripsi
     */
    private function validateEncryptionInput(string $plaintext, string $password):
void
    {
        if (empty($plaintext)) {
            throw new InvalidArgumentException('Plaintext cannot be empty');
        }

        if (strlen($password) < self::MIN_PASSWORD_LENGTH) {
            throw new InvalidArgumentException(
                sprintf('Password must be at least %d characters long',
self::MIN_PASSWORD_LENGTH)
            );
        }
    }

    /**
     * Validasi input untuk dekripsi
     */
    private function validateDecryptionInput(array $encryptedData, string
$password): void
    {
        $requiredFields = ['ciphertext', 'iv', 'salt'];
```

```
        foreach ($requiredFields as $field) {
            if (!isset($encryptedData[$field]) || empty($encryptedData[$field])) {
                throw new InvalidArgumentException("Missing or empty required field: {$field}");
            }
        }

        if (empty($password)) {
            throw new InvalidArgumentException('Password cannot be empty');
        }
    }

    /**
     * Validasi panjang data
     */
    private function validateDataLengths(string $iv, string $salt, string $ciphertext): void
    {
        if (strlen($iv) !== $this->ivLength) {
            throw new InvalidArgumentException('Invalid IV length');
        }

        if (strlen($salt) !== self::SALT_LENGTH) {
            throw new InvalidArgumentException('Invalid salt length');
        }

        if (empty($ciphertext)) {
            throw new InvalidArgumentException('Ciphertext cannot be empty after decoding');
        }
    }

    /**
     * Decode base64 dengan validasi
     */
    private function decodeBase64(string $data, string $fieldName): string
    {
        $decoded = base64_decode($data, true);

        if ($decoded === false) {
            throw new InvalidArgumentException("Invalid base64 format for {$fieldName}");
        }

        return $decoded;
    }

    /**
     * Generate random bytes dengan validasi
     */
```

```
    private function generateRandomBytes(int $length): string
    {
        $bytes = random_bytes($length);

        if ($bytes === false) {
            throw new RuntimeException('Failed to generate random bytes');
        }

        return $bytes;
    }

    /**
     * Get informasi konfigurasi
     */
    public function getConfig(): array
    {
        return [
            'cipher' => self::CIPHER,
            'key_length' => self::KEY_LENGTH,
            'iterations' => self::ITERATIONS,
            'salt_length' => self::SALT_LENGTH,
            'iv_length' => $this->ivLength,
        ];
    }

    /**
     * Verifikasi password tanpa dekripsi penuh (optional)
     */
    public function verifyPassword(string $password, string $storedSalt): bool
    {
        try {
            $salt = $this->decodeBase64($storedSalt, 'salt');
            $this->deriveKey($password, $salt);
            return true;
        } catch (RuntimeException | InvalidArgumentException $e) {
            return false;
        }
    }
}
```

Proses Deskripsi Aes-128

```
<?php

namespace App\Services;

use App\Services\AesPasswordService;
use RuntimeException;

class DecryptService
{
    protected $aesService;
    protected $secretKey;

    public function __construct()
    {
        $this->aesService = new AesPasswordService();
        $this->secretKey = getenv('ENCRYPTION_SECRET_KEY');

        if (!$this->secretKey) {
            throw new RuntimeException('Secret key untuk enkripsi/dekripsi belum
diset di environment.');
        }
    }

    private function validateFields(array $data, array $required)
    {
        foreach ($required as $field) {
            if (empty($data[$field])) {
                throw new RuntimeException("Field $field kosong atau tidak
ditemukan");
            }
        }
    }

    // ========================
    // Transaksi
    // ========================
    public function decryptAlamatTransaksi(array $transaksiItem): string
    {
        // Di DecryptService, sebelum validasi
        //     log_message('debug', 'DEBUG TRANSAKSI ITEM: ' .
print_r($transaksiItem, true));



        try {
            $this->validateFields($transaksiItem, ['alamat_pengiriman', 'iv_alamat',
'salt_alamat']);
            return $this->aesService->decrypt([
                'ciphertext' => $transaksiItem['alamat_pengiriman'],
                'iv' => $transaksiItem['iv_alamat'],
```

```
            'salt' => $transaksiItem['salt_alamat']
        ], $this->secretKey);
    } catch (\Exception $e) {
        // log_message('error', '*Dekripsi* alamat transaksi gagal: ' . $e->getMessage());
        return '[Data tidak dapat di*dekripsi*]';
    }
}

// ========================
// Pengguna
// ========================
public function decryptAlamatPengguna(array $userData): string
{
    try {
        $this->validateFields($userData, ['alamat', 'iv_alamatP', 'salt_alamatP']);
        return $this->aesService->decrypt([
            'ciphertext' => $userData['alamat'],
            'iv' => $userData['iv_alamatP'],
            'salt' => $userData['salt_alamatP']
        ], $this->secretKey);
    } catch (\Exception $e) {
        // log_message('error', '*Dekripsi* alamat pengguna gagal: ' . $e->getMessage());
        return '[Data tidak dapat di*dekripsi*]';
    }
}

public function decryptTeleponPengguna(array $userData): string
{
    try {
        $this->validateFields($userData, ['telepon', 'iv_telepon', 'salt_telepon']);
        return $this->aesService->decrypt([
            'ciphertext' => $userData['telepon'],
            'iv' => $userData['iv_telepon'],
            'salt' => $userData['salt_telepon']
        ], $this->secretKey);
    } catch (\Exception $e) {
        // log_message('error', '*Dekripsi* telepon pengguna gagal: ' . $e->getMessage());
        return '[Data tidak dapat di*dekripsi*]';
    }
}

/**
 * *Dekripsi* lengkap data pengguna
 */
public function decryptPenggunaAll(array $userData): array
{
    $userData['alamat'] = $this->decryptAlamatPengguna($userData);
```

```
        $userData['telepon'] = $this->decryptTeleponPengguna($userData);
        return $userData;
    }

    /**
     * Dekripsi lengkap transaksi + pengguna
     */
    public function decryptTransaksiWithPengguna(array $transaksiItem, array $userData): array
    {
        // log_message('debug', 'DEBUG USER DATA: ' . print_r($userData, true));

        // Dekripsi alamat pengiriman jika tersedia
        if (!empty($userData['alamat_pengiriman']) && !empty($userData['iv_alamat']) && !empty($userData['salt_alamat'])) {
            $transaksiItem['alamat_pengiriman'] = $this->aesService->decrypt([
                'ciphertext' => $userData['alamat_pengiriman'],
                'iv' => $userData['iv_alamat'],
                'salt' => $userData['salt_alamat']
            ], $this->secretKey);
        }

        // Dekripsi data pengguna
        $userData = $this->decryptPenggunaAll($userData);

        // **Tambahkan alamat_pengiriman ke array yang dikembalikan**
        $userData['alamat_pengiriman'] = $transaksiItem['alamat_pengiriman'] ?? null;

        return array_merge($transaksiItem, $userData);
    }
}
```

Proses *Watermark*

```
<?php

namespace App\Services;

use App\Models\M_Watermark;
use CodeIgniter\Files\File;
use RuntimeException;
use InvalidArgumentException;

class WatermarkService
{
    protected $watermarkModel;
    protected $uploadPath;
    protected $whiteThreshold = 230;
    protected $blackThreshold = 80;
    protected $defaultMargin = 20;

    public function __construct()
    {
        $this->watermarkModel = new M_Watermark();
        $this->uploadPath = WRITEPATH . 'uploads/watermark/';

        $this->validatePhpVersion();
        $this->ensureUploadDirectoryExists();
    }

    /**
     * Apply watermark to target image
     */
    public function applyToImage(string $targetImagePath): bool
    {
        try {
            $watermark = $this->watermarkModel->getActiveWatermark();

            if (!$watermark || !$this->shouldApplyWatermark()) {
                return false;
            }

            return $this->processWatermark($targetImagePath, $watermark);
        } catch (\Exception $e) {
            log_message('error', 'Watermark Error: ' . $e->getMessage());
            return false;
        }
    }

    /**
     * Check if watermark should be applied
     */
    protected function shouldApplyWatermark(): bool
```

```
    {
        return true;
    }

    /**
     * Main watermark processing logic
     */
    protected function processWatermark(string $targetImagePath, array $watermarkConfig): bool
    {
        $watermarkPath = $this->uploadPath . $watermarkConfig['path_gambar'];

        $this->validateFileExistence($targetImagePath, $watermarkPath);

        $mainImage = $this->createImageResource($targetImagePath);
        $wmImage = imagecreatefrompng($watermarkPath);

        $this->validateImageResources($mainImage, $wmImage);

        // Set alpha blending and save alpha for main image
        imagealphablending($mainImage, true);
        imagesavealpha($mainImage, true);

        // Set alpha blending and save alpha for watermark image
        imagealphablending($wmImage, true);
        imagesavealpha($wmImage, true);

        // Skip removeBlackWhiteBackground if watermark is already transparent
        if (!$this->hasTransparency($wmImage)) {
            $this->removeBlackWhiteBackground($wmImage);
        }

        $resizedWM = $this->resizeWatermark($wmImage, $watermarkConfig);
        $position = $this->calculatePosition($mainImage, $resizedWM, $watermarkConfig);

        $this->mergeWatermark($mainImage, $resizedWM, $position, $watermarkConfig);
        $this->saveImage($mainImage, $targetImagePath);

        $this->cleanupResources($mainImage, $wmImage, $resizedWM);

        return true;
    }

    /**
     * Check if image has transparency
     */
    protected function hasTransparency($image): bool
    {
```

```
        if (!$this->isGdImage($image)) {
            return false;
        }

        $width = imagesx($image);
        $height = imagesy($image);

        for ($x = 0; $x < $width; $x++) {
            for ($y = 0; $y < $height; $y++) {
                $colorIndex = imagecolorat($image, $x, $y);
                $colorInfo = imagecolorsforindex($image, $colorIndex);

                if ($colorInfo['alpha'] > 0) {
                    return true;
                }
            }
        }

        return false;
    }

    /**
     * Remove white or black background automatically
     */
    protected function removeBlackWhiteBackground($image): bool
    {
        if (!$this->isGdImage($image)) {
            return false;
        }

        $width = imagesx($image);
        $height = imagesy($image);
        $removed = false;

        for ($x = 0; $x < $width; $x++) {
            for ($y = 0; $y < $height; $y++) {
                $pixelColor = imagecolorat($image, $x, $y);
                $r = ($pixelColor >> 16) & 0xFF;
                $g = ($pixelColor >> 8) & 0xFF;
                $b = $pixelColor & 0xFF;

                $isWhite = ($r >= $this->whiteThreshold && $g >= $this->whiteThreshold && $b >= $this->whiteThreshold);
                $isBlack = ($r <= $this->blackThreshold && $g <= $this->blackThreshold && $b <= $this->blackThreshold);

                if ($isWhite || $isBlack) {
                    $transparent = imagecolorallocatealpha($image, 0, 0, 0, 127);
                    imagesetpixel($image, $x, $y, $transparent);
                    $removed = true;
```

```
            }
        }
    }

    return $removed;
}

/**
 * Type-safe image resource check
 */
protected function isGdImage($resource): bool
{
    if (PHP_MAJOR_VERSION >= 8) {
        return $resource instanceof \GdImage;
    }

    return is_resource($resource) && get_resource_type($resource) === 'gd';
}

/**
 * Create image resource based on file type
 */
protected function createImageResource(string $imagePath)
{
    if (!file_exists($imagePath)) {
        throw new RuntimeException('Image file not found: ' . $imagePath);
    }

    $extension = strtolower(pathinfo($imagePath, PATHINFO_EXTENSION));
    $image = false;

    switch ($extension) {
        case 'jpg':
        case 'jpeg':
            $image = imagecreatefromjpeg($imagePath);
            break;
        case 'png':
            $image = imagecreatefrompng($imagePath);
            // Enable alpha blending and save alpha for PNG images
            imagealphablending($image, true);
            imagesavealpha($image, true);
            break;
        case 'gif':
            $image = imagecreatefromgif($imagePath);
            // Enable alpha blending for GIF images
            imagealphablending($image, true);
            break;
        case 'webp':
            $image = function_exists('imagecreatefromwebp') ?
```

```
imagecreatefrom*webp*($imagePath) : false;
            if ($image && $this->isGdImage($image)) {
                // Enable alpha blending and save alpha for *Web*P images
                imagealphablending($image, true);
                imagesavealpha($image, true);
            }
            break;
        default:
            throw new RuntimeException('Unsupported image format: ' .
$extension);
    }

    if (!$this->isGdImage($image)) {
        throw new RuntimeException('Failed to create image resource from: ' .
$imagePath);
    }

    return $image;
}

/**
 * Resize *watermark* based on configuration
 */
protected function resize*Watermark*($wmImage, array $config)
{
    if (!$this->isGdImage($wmImage)) {
        throw new InvalidArgumentException('Invalid *watermark* image
resource');
    }

    $wmWidth = imagesx($wmImage);
    $wmHeight = imagesy($wmImage);
    $scale = $this->getScaleFactor($wmWidth, $config);

    $newW = (int)($wmWidth * $scale);
    $newH = (int)($wmHeight * $scale);

    $resizedWM = imagecreatetruecolor($newW, $newH);

    if (!$this->isGdImage($resizedWM)) {
        throw new RuntimeException('Failed to create resized *watermark*
resource');
    }

    // Enable alpha blending and save alpha for resized *watermark*
    imagealphablending($resizedWM, false);
    $transparent = imagecolorallocatealpha($resizedWM, 0, 0, 0, 127);
    imagefill($resizedWM, 0, 0, $transparent);
    imagesavealpha($resizedWM, true);
```

```
        imagecopyresampled($resizedWM, $wmImage, 0, 0, 0, 0, $newW, $newH,
$wmWidth, $wmHeight);

        return $resizedWM;
    }

    /**
     * Calculate scale factor
     */
    protected function getScaleFactor(int $wmWidth, array $config): float
    {
        switch ($config['ukuran']) {
            case 'kecil':
                return 0.3;
            case 'besar':
                return 0.8;
            case 'custom':
                return  $config['ukuran_custom']  >  0  ?  $config['ukuran_custom']  /
$wmWidth : 0.5;
            default:
                return 0.5;
        }
    }

    /**
     * Calculate watermark position
     */
    protected function calculatePosition($mainImage, $watermark, array $config):
array
    {
        if (!$this->isGdImage($mainImage) || !$this->isGdImage($watermark)) {
            throw new InvalidArgumentException('Invalid image resources for
position calculation');
        }

        $imgWidth = imagesx($mainImage);
        $imgHeight = imagesy($mainImage);
        $wmWidth = imagesx($watermark);
        $wmHeight = imagesy($watermark);

        $margin = $this->defaultMargin;

        $positions = [
            'atas-kiri'     => ['x' => $margin, 'y' => $margin],
            'atas-tengah'       => ['x' =>  (int)(($imgWidth  -  $wmWidth)  /  2), 'y' =>
$margin],
            'atas-kanan'        => ['x' =>  $imgWidth  -  $wmWidth  -  $margin, 'y' =>
$margin],
            'tengah-kiri'    => ['x' => $margin, 'y' => (int)(($imgHeight - $wmHeight)
/ 2)],
```

```
            'tengah'          => ['x' => (int)(($imgWidth - $wmWidth) / 2), 'y' => (int)(($imgHeight - $wmHeight) / 2)],
            'tengah-kanan'    => ['x' => $imgWidth - $wmWidth - $margin, 'y' => (int)(($imgHeight - $wmHeight) / 2)],
            'bawah-kiri'      => ['x' => $margin, 'y' => $imgHeight - $wmHeight - $margin],
            'bawah-tengah'    => ['x' => (int)(($imgWidth - $wmWidth) / 2), 'y' => $imgHeight - $wmHeight - $margin],
            'bawah-kanan'     => ['x' => $imgWidth - $wmWidth - $margin, 'y' => $imgHeight - $wmHeight - $margin]
        ];

        return $positions[$config['posisi']] ?? ['x' => $margin, 'y' => $margin];
    }

    /**
     * Helper function to copy and merge images with alpha transparency
     */
    protected function imagecopymerge_alpha($dst_im, $src_im, $dst_x, $dst_y, $src_x, $src_y, $src_w, $src_h, $pct)
    {
        // Create a cut resource
        $cut = imagecreatetruecolor($src_w, $src_h);

        // Copy relevant section from background to the cut resource
        imagecopy($cut, $dst_im, 0, 0, $dst_x, $dst_y, $src_w, $src_h);

        // Copy relevant section from *watermark* to the cut resource
        imagecopy($cut, $src_im, 0, 0, $src_x, $src_y, $src_w, $src_h);

        // Insert cut resource to destination image
        imagecopymerge($dst_im, $cut, $dst_x, $dst_y, 0, 0, $src_w, $src_h, $pct);

        // Clean up
        imagedestroy($cut);
    }

    /**
     * Merge *watermark* with main image
     */
    protected function merge*Watermark*($mainImage, $*watermark*, array $position, array $config): void
    {
        if (!$this->isGdImage($mainImage) || !$this->isGdImage($*watermark*)) {
            throw new InvalidArgumentException('Invalid image resources for merging');
        }

        $opacity = max(10, min(100, (int)$config['transparansi']));
        $wmWidth = imagesx($*watermark*);
```

```
        $wmHeight = imagesy($watermark);

        // If opacity is 100%, use imagecopy for better transparency handling
        if ($opacity == 100) {
            imagecopy(
                $mainImage,
                $watermark,
                $position['x'],
                $position['y'],
                0,
                0,
                $wmWidth,
                $wmHeight
            );
        } else {
            // Use our helper function for partial transparency
            $this->imagecopymerge_alpha(
                $mainImage,
                $watermark,
                $position['x'],
                $position['y'],
                0,
                0,
                $wmWidth,
                $wmHeight,
                $opacity
            );
        }
    }

    /**
     * Save image based on format
     */
    protected function saveImage($image, string $path): void
    {
        if (!$this->isGdImage($image)) {
            throw new InvalidArgumentException('Invalid image resource for saving');
        }

        $extension = strtolower(pathinfo($path, PATHINFO_EXTENSION));

        // Preserve alpha channel for supported formats
        if ($extension === 'png' || $extension === 'webp') {
            imagealphablending($image, false);
            imagesavealpha($image, true);
        }

        $saveMethods = [
            'jpg'  => fn() => imagejpeg($image, $path, 90),
```

```
            'jpeg' => fn() => imagejpeg($image, $path, 90),
            'png'  => fn() => imagepng($image, $path, 9),
            'gif'  => fn() => imagegif($image, $path),
            'webp' => fn() => function_exists('imagewebp') ? imagewebp($image,
$path, 90) : imagejpeg($image, $path, 90)
        ];

        if (isset($saveMethods[$extension])) {
            $saveMethods[$extension]();
        } else {
            imagejpeg($image, $path, 90);
        }
    }

    /**
     * Cleanup image resources
     */
    protected function cleanupResources(...$resources): void
    {
        foreach ($resources as $resource) {
            if ($this->isGdImage($resource)) {
                imagedestroy($resource);
            }
        }
    }

    /**
     * Get active watermark configuration
     */
    public function getActiveWatermark(): ?array
    {
        return $this->watermarkModel->getActiveWatermark();
    }

    /**
     * Check if watermark is active
     */
    public function isWatermarkActive(): bool
    {
        $watermark = $this->getActiveWatermark();
        return !empty($watermark) && file_exists($this->uploadPath .
$watermark['path_gambar']);
    }

    /**
     * Validate PHP version compatibility
     */
    protected function validatePhpVersion(): void
    {
        if (PHP_MAJOR_VERSION < 7 || (PHP_MAJOR_VERSION === 7 &&
```

```
PHP_MINOR_VERSION < 4)) {
            throw new RuntimeException('PHP 7.4 or higher is required for GD operations');
        }
    }

    /**
     * Ensure upload directory exists
     */
    protected function ensureUploadDirectoryExists(): void
    {
        if (!is_dir($this->uploadPath)) {
            mkdir($this->uploadPath, 0777, true);
        }
    }


    /**
     * Validate file existence
     */
    protected function validateFileExistence(string $targetImagePath, string $watermarkPath): void
    {
        if (!file_exists($watermarkPath) || !file_exists($targetImagePath)) {
            throw new RuntimeException('Watermark or target image not found');
        }
    }

    /**
     * Validate image resources
     */
    protected function validateImageResources($mainImage, $wmImage): void
    {
        if (!$this->isGdImage($mainImage) || !$this->isGdImage($wmImage)) {
            throw new RuntimeException('Failed to create image resources');
        }
    }
}
```

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

BERITA ACARA
UJIAN TUGAS AKHIR

Periode : November 2025

Pada hari ini Selasa, 18 November 2025 Pukul 09.00 WIB sampai dengan ..... WIB telah dilaksanakan Ujian Tugas Akhir mahasiswa atas nama :

Nama Mahasiswa	:	Muhammad Isnaini
N I M	:	C2155201110
Program Studi	:	Teknik Informatika
Tahun Angkatan	:	2021
Judul TA	:	Implementasi Keamanan Penjualan Baju Adat Berbasis Web menggunakan Watermarking dan Enkripsi AES

Dosen Penguji :

	Nama		Nilai	Tanda Tangan
1.	Amaya Andri Damaini, S.Kom., M.T	=	.....	(.....)
2.	Elok Faiqotul Himmah, S.Si., M.Sc	=	.....	(.....)
3.	Elia Zakharia, M.T	=	.....	(.....)
4.	Ir. Abdul Hadi, S.T., M.Kom	=	.....	(.....)
5.	Rudini, M.Pd	=	.....	(.....)

Hasil Ujian : **LULUS** / ~~**TIDAK LULUS** *~~) NILAI = ...80,41.....

Revisi (~~Tidak Ada~~ / Minor / ~~Mayor~~ *)

Catatan Penting :
1. Lama Perbaikan : .....14..... hari
2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian.
3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru.

Demikian Berita Acara ini dibuat, agar dapat digunakan sebagaimana semestinya.

Palangka Raya, Selasa, 18 November 2025

Ketua Penguji,

Amaya Andri Damaini, S.Kom., M.T
NIK. 199309042018001

Mahasiswa,

Muhammad Isnaini
NIM. C2155201110

Mengetahui :
Ketua Program Studi Teknik Informatika,

Ir. Abdul Hadi, S.T., M.Kom
NIK. 198505102021101

Tembusan:
Mahasiswa yang bersangkutan.
Berita acara ini diberikan setelah menyelesaikan semua revisi dari semua penguji
**) Coret yang tidak perlu*

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS

No.50/STMIK-3.C.2/KP/II/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama- nama tersebut di bawah ini :

1. Nama : Abdul Hadi, S.T., M.Kom.
   NIK : 198505102021101
   Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Rudini, M.Pd.
   NIK : 198709172015105
   Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

Untuk membimbing Tugas Akhir mahasiswa :

Nama	:	Muhammad Isnaini
NIM	:	C2155201110
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Daftar	:	28 Agustus 2024
Judul Tugas Akhir	:	Implementasi Keamanan Aplikasi Penjualan Baju Adat Pada Toko Bawi Lamus Palangka Raya Menggunakan Algoritma AES

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 21 Februari 2025
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :
1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip

Dipindai dengan CamScanner

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI TUGAS AKHIR

Nama Mahasiswa : Muhammad Isnaini
NIM : C2155201110
Tanggal Persetujuan Judul : .....
Judul Tugas Akhir : Implementasi Kecerdasan Aplikasi Pinjaman Baju Adat Menggunakan Algoritma AES Pada Baju Lamas Palangkaraya

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	23/12/2024	24/12/2024	Memperbaiki Latar belakang dan Penulisan	
	20/02/2025	03/03/2025	Memperbaiki Urgensi fenomena terjadi Pada Latar belakang, Penambahan Kata pengantar dan Konsep Memilih Metode AES	
	26/05/2025	28/05/2025	Memperbaiki lengkap daftar pustaka dan Memperbaiki daftar Isi	
	02/6/2025		lampiran tambahkan dokumentasi wawancara dan foto	
	03/6/2025		Acc Seminar	
	14/01/2025	16/01/2025	Perbaikan Penulisan Rata kanan kiri Sesuai Pedoman	
	27/05/2025	30/05/2025	Perbaikan daftar Isi Menyesuaikan Pedoman	
	02-06/2025		Pembenaran daftar Isi dan Bab 2 tentang Kajian Umum teori dan kajian pustaka	
	03/06/2025		Perbaikan Margin dan nomor halaman tertera Rapih	

Dipindai dengan CamScanner

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) PALANGKARAYA

Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI TUGAS AKHIR

Nama Mahasiswa : Muhammad Isnaini

NIM : C2155201110

Tanggal Persetujuan Judul : 08-01-2025

Judul Tugas Akhir : Implementasi Keamanan Penjualan baju Aplikasi berbasis web menggunakan Waterfall dan AES

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
	10/10/2025	11/10/2025	Penyesuaian antara Daftar Isi Sinkronisasi dengan Hasil dan Pembahasan	
	13/10/2025	14/10/2025	Penyesuai ABStrak Bahasa Inggris dan Mobile SC	
	15/10/2025	16/10/2025	Implementasi keamanan TA dan sistem	
	17/10/2025	18/10/2025	Jarak nama dan nim spasi single, bagian nim berapa garis bawah	
	20/10/2025	21/10/2025	Penyesuaian lanjut Keamanan Sistem TA	
	22/10/2025	22/10/2025	oke Siap	

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS
PENGUJI SEMINAR PROPOSAL TUGAS AKHIR

No.131/STMIK-3.C.2/AU/VI/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama-nama berikut :

1. Nama : Elia Zakharia, M.T.
   NIK : 199205262016104
   Sebagai Ketua
2. Nama : Ir. Abdul Hadi, S.T., M.Kom
   NIK : 198505102021101
   Sebagai Sekretaris
3. Nama : Rudini, M.Pd
   NIK : 198709172015105
   Sebagai Anggota

Untuk menguji seminar proposal tugas akhir mahasiswa :

Nama Mahasiswa	:	Muhammad Isnaini
N I M	:	C2155201110
Hari / Tanggal Ujian	:	Jumat, 13 Juni 2025
Waktu	:	08.30 WIB
Ruangan	:	Seminar 1
Judul TA	:	IMPLEMENTASI KEAMANAN APLIKASI PENJUALAN BAJU ADAT MENGGUNAKAN ALGORITMA AES PADA BAWI LAMUS PALANGKARAYA

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 10 Juni 2025

Ketua Program Studi Teknik Informatika,

10/6/2025

Ir. ABDUL HADI, S.T., M.Kom
NIK. 198505102021101

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip

CS Dipindai dengan CamScanner

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN SEMINAR PROPOSAL TUGAS AKHIR

Nama Mahasiswa : Muhammad Ishaini
NIM : C2155201110
Program Studi : Teknik Informatika

No.	Hari/ Tanggal	Judul	Mahasiswa Penyaji	Nama Tim Dosen	Tanda Tangan
1	Jum'at, 21 Feb, 2025	Diagnosa tingkat kecemasan pentakit menggunakan fuzzy sugeno	Nopri Aldani Saputra	1 Hotmian Sitohang, M, Kom 2 Dr. Maria Ulfa Ningsih, S.Kom M.Cs.	
2	Jum'at 27-11-2024	Prototipe sistem penghitung otomatis jumlah jemaat berbasis IOT pada gereja evanggelion	Imanuel Rudius Tinbaca	3 Rusmi, M.Pd Hercules, S.Kom Amana, A.O.S.Kom Hotmian, S.S.Kom	
3	Sabtu 23-11-2024	Analisis jaringan Hospot dengan parameter Quality Of Struice (QoS) pada kantor wilayah kementerian agama provinsi kalimantan tengah	Tauhker rohman	Ir. Hj. Siti Mariyah Sucisiawati, S.Kom Rusmi, M.Pd	
4	Sabtu 23-11-2024	Implementasi teknologi IOT pada kebakaran Ucok Lili Berbasis Web untuk monitoring kebakaran api	Ahmad Husaini	Somomi, ST. M.Kom Hercules S.Kom Lili Kusdiarsa, M.Kom	
5	Jum'at 20-12-2024	Prototipe pendeteksi hujan pada atap otomatis untuk lingkungan rumah menggunakan ESP 8266 dan Blynk	STEVIN WINATA	ELIA Zacharia, M.T Hercules, S.Kom Dr. Mura Widiarsih, S.Kom	

Palangka Raya, 09-08-2025
Mahasiswa ybs,

Muhammad Ishaini

Keterangan :
- *) Coret yang tidak perlu
- Harap kartu jangan sampai hilang, digunakan sebagai syarat seminar
- Minimal 5 (lima) kali mengikuti seminar

CS Dipindai dengan CamScanner

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

Nomor : 775/STMIK-C.1.1.Ak/X/2025
Lampiran : -
Perihal : Permohonan Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir

Kepada
Yth. **Owner Bawi Lamus Palangkaraya**

Jl. Beliang, Palangkaraya
Provinsi Kalimantan Tengah

Dengan hormat,

Sehubungan dengan penyusunan Tugas Akhir mahasiswa sebagai persyaratan kelulusan Program Studi Teknik Informatika (S1) pada STMIK Palangkaraya, maka dengan ini kami sampaikan permohonan izin penelitian dan pengumpulan data bagi mahasiswa kami berikut:

Nama	:	MUHAMMAD ISNAINI
NIM	:	C2155201110
Prodi (Jenjang)	:	Teknik Informatika (S1)
Thn. Akad. (Semester)	:	2025/2026 (9)
Lama Penelitian	:	21 Oktober 2025 s.d 21 November 2025
Tempat Penelitian	:	Toko Baju Adat Bawi Lamus Palangkaraya

Dengan judul Tugas Akhir:

Implementasi Keamanan Penjualan Baju Adat Berbasis Web Menggunakan Watermarking Dan AES

Adapun ketentuan dan aturan pemberian informasi dan data yang diperlukan dalam penelitian tersebut menyesuaikan dengan ketentuan/peraturan pada instansi Bapak/Ibu.

Demikian permohonan ini disampaikan, atas perhatian dan kerja samanya diucapkan terima kasih.

Palangka Raya, 21 Oktober 2025

Ketua,

Ir. Hj. Siti Maryamah, MM.
NIK. 196902021995004

LEMBAR BERITA ACARA PENGUJIAN BLACK BOX

Uji Coba " *IMPLEMENTASI KEAMANAN PENJUALAN BAJU ADAT BERBASIS WEB MENGGUNAKAN WATERMARKING DAN ENKRIPSI AES* "

Nama : Muhammad Isnaini
NIM : C2155201110
Jurusan : Teknik Informatika

Jika hasil pengujian sesuai dengan yang diharapkan maka kolom hasil pengujian diisi dengan simbol "centang" (√), jika tidak maka isi dengan simbol "strip"(-).

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Menampilkan halaman login admin.	Sistem menampilkan halaman login admin.	√
Memasukan E-mail dan password jika salah.	Sistem menampilkan pesan login gagal jika E-mail atau password salah batas percobaan 5 kali.	√
Memasukan E-mail dan password jika benar.	Sistem langsung mengarahkan ke halaman dashboard admin.	√
Masuk kedalam halaman dashboard admin.	Sistem akan menampilkan halaman dashboard admin.	√
Memilih halaman transaksi.	Sistem akan menampilkan halaman transaksi.	√
Memilih detail transaksi	Sistem akan menampilkan data detail transaksi berdasarkan id yang dipilih.	√
Memilih status pesanan.	Sistem akan menampilkan option untuk status pesanan dan dapat berubah di halaman pengguna juga agar dapat melacak status pesanan.	√

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih deskripsi dan memasukan password untuk deskripsi jika salah.	Sistem akan menampilkan masukan password dan jika salah ada tampilan password salah.	✓
Memilih deskripsi dan memasukan password untuk deskripsi jika benar.	Sistem akan menampilkan masukan password dan jika benar ada tampilan data berhasil dideskripsi.	✓
Memilih Cetak transaksi berdasarkan id transaksi.	Sistem akan menampilkan halaman untuk melakukan cetak pada data transaksi yang dipilih.	✓
Memilih hapus transaksi.	Sistem akan melakukan penghapusan berdasarkan id transaksi yang dipilih dan akan menampilkan pesan data transaksi berhasil dihapus.	✓
Memilih halaman produk.	Sistem akan memunculkan dropdown yang ada data produk dan data kategori.	✓
Memilih data produk	Sistem akan menampilkan halaman produk yang dapat melakukan CRUD pada data produk.	✓
Melakukan penambahan data produk pada halaman data produk jika salah dan tidak sesuai dengan validasi.	Sistem akan memberikan form untuk mengisi data produk kemudian akan menampilkan pesan error pada bagian yang tidak sesuai dengan validasi.	✓
Melakukan penambahan data produk pada halaman data produk jika benar.	Sistem akan memberikan form untuk mengisi data produk kemudian akan menampilkan pesan sukses kemudian data produk berhasil ditambahkan.	✓
Memilih detail produk.	Sistem akan menampilkan data produk berdasarkan id produk yang dipilih.	✓
Memilih update/pembaharuan pada data produk jika gagal memasukan data yang tidak sesuai dengan validasi.	Sistem akan memberikan form untuk melakukan update/pembaharuan jika gagal maka akan menampilkan pesan	✓

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Melakukan Update/Pembaharuan pada data kategori jika berhasil.	Sistem akan menampilkan form untuk melakukan update/pembaharuan pada data kategori jika sesuai dan validasi sistem benar maka akan menampilkan pesan data kategori berhasil diperbaharui.	✓
Melakukan hapus data kategori berdasarkan id data kategori.	Sistem akan menampilkan pilihan apakah akan menghapus data kategori, jika ya maka data kategori akan terhapus dan jika batal maka tidak terhapus.	✓
Memilih halaman konfigurasi pada dashboard.	Sistem akan menampilkan dropdown atau pilihan menu antara pengaturan akun dan logo watermark.	✓
Memilih halaman pengaturan akun	Sistem akan menampilkan form untuk melakukan perubahan password pada akun admin.	✓
Memilih halaman pengaturan logo watermark.	Sistem akan menampilkan dan memberikan form watermark.	✓
Melakukan penambahan logo watermark jika gagal.	Sistem akan menampilkan pesan gagal yang tidak sesuai dengan validasi sistem.	✓
Melakukan penambahan logo watermark jika berhasil.	Sistem akan menampilkan pesan sukses jika sesuai dengan validasi sistem.	✓
Memilih Logout	Sistem akan menampilkan pilihan untuk keluar, apakah ya atau batal. Jika ya maka akan mengarahkan ke halaman login admin kembali dan session berakhir jika batal maka tidak akan keluar dari halaman dashboard.	✓

SKENARIO UJI	HASIL YANG DIHARAPKAN	KESIMPULAN
Memilih konfirmasi pesanan.	Sistem akan mengarahkan ke pilihan , periksa kembali dan ya, konfirmasi pesanan. Dan akan mengarahkan kembali ke halaman keranjang untuk melakukan pembelanjaan lagi. Pada bagian ini data transaksi pelanggan juga akan di enkripsi oleh sistem menggunakan AES-128	✓
Melakukan pembayaran sesuai dengan yang dipilih.	Sistem akan menampilkan halaman payment gateway yang sesuai dengan tipe pembayaran yang dipilih oleh user.	✓
Memilih halaman riwayat belanja.	Sistem akan menampilkan halaman dari riwayat belanja.	✓
Memilih detail riwayat belanja.	Sistem akan menampilkan detail dari riwayat belanja.	✓
Memilih logout.	Sistem akan menghapus session dan mengarahkan ke halaman login pelanggan lagi dan menampilkan pesan sukses.	✓

Palangka Raya, 16 Oktober 2025
Dosen Penguji,

Amaya Andri Damaini, S.Kom., M.T.
NIK.199304092016006



The image part with relationship ID rId105 was not found in the file.


WEDDING ADAT DAYAK
Bawi
Lamus
BAWILAMUS
Wedding Adat Dayak


WEDDING ADAT DAYAK
Bawi
Lamus
BAWILAMUS
Wedding Adat Dayak