
Academia Open



By Universitas Muhammadiyah Sidoarjo

Academia Open

Vol. 11 No. 1 (2026): June
DOI: 10.21070/acopen.11.2026.14411

Table Of Contents

Journal Cover	1
Author[s] Statement	3
Editorial Team	4
Article information	5
Check this article update (crossmark)	5
Check this article impact	5
Cite this article.....	5
Title page	6
Article Title	6
Author information	6
Abstract	6
Article content	7

Originality Statement

The author[s] declare that this article is their own work and to the best of their knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the published of any other published materials, except where due acknowledgement is made in the article. Any contribution made to the research by others, with whom author[s] have work, is explicitly acknowledged in the article.

Conflict of Interest Statement

The author[s] declare that this article was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright Statement

Copyright © Author(s). This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

Academia Open

Vol. 11 No. 1 (2026): June
DOI: 10.21070/acopen.11.2026.14411

EDITORIAL TEAM

Editor in Chief

Mochammad Tanzil Multazam, Universitas Muhammadiyah Sidoarjo, Indonesia

Managing Editor

Bobur Sobirov, Samarkand Institute of Economics and Service, Uzbekistan

Editors

Fika Megawati, Universitas Muhammadiyah Sidoarjo, Indonesia

Mahardika Darmawan Kusuma Wardana, Universitas Muhammadiyah Sidoarjo, Indonesia

Wiwit Wahyu Wijayanti, Universitas Muhammadiyah Sidoarjo, Indonesia

Farkhod Abdurakhmonov, Silk Road International Tourism University, Uzbekistan

Dr. Hindarto, Universitas Muhammadiyah Sidoarjo, Indonesia

Evi Rinata, Universitas Muhammadiyah Sidoarjo, Indonesia

M Faisal Amir, Universitas Muhammadiyah Sidoarjo, Indonesia

Dr. Hana Catur Wahyuni, Universitas Muhammadiyah Sidoarjo, Indonesia

Complete list of editorial team ([link](#))

Complete list of indexing services for this journal ([link](#))

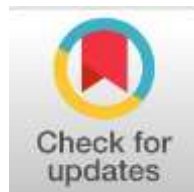
How to submit to this journal ([link](#))

Academia Open

Vol. 11 No. 1 (2026): June
DOI: 10.21070/acopen.11.2026.14411

Article information

Check this article update (crossmark)



Check this article impact (*)



Save this article to Mendeley



(*) Time for indexing process is various, depends on indexing database platform

Laravel Integration Validates E Commerce and Workshop Services: Integrasi Laravel Memvalidasi Layanan E Commerce dan Workshop

Alfi Permana Putra, alfipermana1@gmail.com

Program Studi Informatika, Universitas Amikom Purwokerto, Indonesia

Andi Dwi Riyanto, andi@amikompurwokerto.ac.id (*)

Departemen: Sistem Informasi, Universitas Amikom Purwokerto, Indonesia

Argiyan Dwi Pritama, argiyandwi@amikompurwokerto.ac.id

Departemen: Sistem Informasi, Universitas Amikom Purwokerto, Indonesia

(*) Corresponding author

Abstract

General Background: Web-based digitalization supports micro, small, and medium enterprises in organizing transactions, customer services, and business data through more structured systems. **Specific Background:** The studied batik enterprise still used manual transaction recording and workshop booking management, causing inefficiencies in sales data, payment confirmation, shipping cost calculation, and order administration. **Knowledge Gap:** Previous studies generally discussed e-commerce systems and booking systems separately, while integrated backend development for both services in one platform remained less emphasized. **Aims:** This study developed a Laravel-based backend that integrates e-commerce and workshop booking services using a Waterfall development model. **Results:** The backend was built with MVC architecture, MySQL database, Tripay payment gateway, and RajaOngkir API to support authentication, product management, shopping cart, checkout, onlinepayment, automatic shipping costcalculation, workshop booking, paymentwebhook, and admin dashboard. Black-box testing on 12 main scenarios showed that all scenarios were successful, achieving a 100% functional success rate. **Novelty:** The system combines product transactions, workshop reservations, digital payment processing, shipping cost automation, webhook synchronization, and administrative monitoring in one backend architecture. **Implications:** The developed backend provides an integrated digitalservice foundation that reduces manualdependency and supports more efficient product transactions, workshop booking, and administrative management.

Highlights

- Twelve black-box scenarios produced a 100% successful functional outcome.
- Tripay and RajaOngkir supported automated payment and shipping cost processes.
- Admin dashboard centralized transaction monitoring and reservation management.

Keywords

Laravel Backend; E-Commerce; Workshop Booking; Payment Gateway; Black-Box Testing

Published date: 2026-05-28

Pendahuluan

Perkembangan teknologi informasi, khususnya teknologi web, telah memberikan dampak signifikan terhadap pertumbuhan sektor Usaha Mikro, Kecil, dan Menengah (UMKM). Digitalisasi melalui sistem berbasis web memungkinkan UMKM untuk memperluas jangkauan pasar, meningkatkan efisiensi operasional, dan memfasilitasi transaksi secara online. Penerapan e-commerce telah terbukti meningkatkan efektivitas dan efisiensi bisnis dengan mengurangi biaya operasional, mempercepat proses transaksi, dan meningkatkan kepuasan pelanggan. Selain itu, digitalisasi memainkan peran penting dalam meningkatkan daya saing UMKM melalui optimalisasi proses bisnis dan pengelolaan data yang lebih terstruktur.

Meskipun demikian, masih banyak UMKM yang belum mengoptimalkan pemanfaatan teknologi digital dalam operasional bisnisnya, terutama dalam aspek pengelolaan transaksi dan layanan pelanggan. Salah satu contohnya adalah Batik Giri Alam yang masih menggunakan sistem manual dalam proses transaksi, pencatatan data, serta pengelolaan layanan pelanggan, sehingga menimbulkan berbagai permasalahan seperti tingginya potensi kesalahan pencatatan, keterbatasan jangkauan pemasaran, serta kurang efisiennya proses pelayanan kepada pelanggan. Kondisi ini sejalan dengan penelitian yang menunjukkan bahwa banyak UMKM masih menghadapi kendala dalam sistem pencatatan transaksi yang dilakukan secara manual sehingga menyebabkan ketidakefisienan dan kesulitan dalam pengelolaan data usaha. Selain itu, kebutuhan sistem tidak hanya terbatas pada penjualan produk, tetapi juga mencakup layanan tambahan seperti booking workshop yang menjadi bagian dari strategi bisnis dalam meningkatkan interaksi dengan pelanggan. Oleh karena itu, diperlukan suatu sistem backend yang mampu mengelola data, logika bisnis, serta integrasi antara fitur e-commerce dan booking workshop secara efektif dan terstruktur.

Berbagai penelitian terdahulu telah mengkaji pengembangan sistem e-commerce dan sistem booking berbasis web. Penelitian oleh menunjukkan bahwa penggunaan framework Laravel dalam pembangunan sistem e-commerce mampu meningkatkan efisiensi pengelolaan produk dan transaksi serta mempermudah proses pengembangan sistem. Sebuah studi lain oleh juga mengungkapkan bahwa pengembangan platform e-commerce berbasis Laravel dengan metode Waterfall menghasilkan sistem yang efektif dan mampu meningkatkan kinerja pemasaran UMKM. Di sisi lain, sebuah studi tentang sistem booking online oleh menunjukkan bahwa sistem booking berbasis web dapat meningkatkan efisiensi layanan serta mengurangi kesalahan pencatatan. Selain itu, kajian literatur oleh menyatakan bahwa penggunaan framework modern seperti Laravel memberikan keunggulan dari hal keamanan, kinerja, serta kemudahan pengembangan aplikasi berbasis web. Meskipun demikian, sebagian besar penelitian tersebut masih berfokus pada pengembangan sistem e-commerce dan sistem booking secara terpisah, serta belum menekankan pada pengembangan backend sebagai inti sistem yang mengintegrasikan kedua layanan tersebut dalam satu platform yang terpadu.

Berdasarkan kajian tersebut, teridentifikasi adanya gap penelitian berupa belum optimalnya pengembangan backend yang mampu mengintegrasikan layanan e-commerce dan booking workshop dalam satu sistem yang terstruktur. Pada kenyataannya, integrasi kedua layanan ini sangat penting untuk meningkatkan efisiensi operasional dan memberikan pengalaman pengguna yang lebih baik. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan backend situs web berbasis Laravel yang mampu mendukung layanan e-commerce dan booking workshop terintegrasi di Batik Giri Alam. Dengan sistem ini, diharapkan efisiensi pengelolaan data akan meningkat, proses transaksi akan disederhanakan, dan pengembangan bisnis digital akan didukung. Keunikan penelitian ini terletak pada pengembangan backend terintegrasi yang tidak hanya berfokus pada transaksi e-commerce, tetapi juga menggabungkan layanan pemesanan workshop, integrasi pembayaran online melalui Tripay, perhitungan biaya pengiriman otomatis menggunakan RajaOngkir, dan dashboard admin ke dalam satu sistem. Integrasi ini membedakannya dari penelitian sebelumnya, yang umumnya masih membahas sistem e-commerce dan sistem booking secara terpisah.

Metode

A. Metode Pengembangan

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Waterfall. Metode waterfall adalah metode yang dikembangkan untuk pengembangan perangkat lunak, membuat perangkat lunak. Metode berkembang secara sistematis dari satu tahap ke tahap lain dalam model seperti air terjun. Metode waterfall sering juga disebut model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle).

Metode ini dipilih karena memiliki tahapan yang terstruktur dan sistematis sehingga memudahkan dalam proses pengembangan perangkat lunak. Berikut pada gambar 1 visualisasi tahapan metode *waterfall*:

1. Analisis Kebutuhan

Tahapan pertama adalah analisis kebutuhan, yang dilakukan untuk mengidentifikasi kebutuhan sistem berdasarkan permasalahan yang ada pada Batik Giri Alam, baik dari sisi pengguna maupun sistem. Pada tahap ini dilakukan pengumpulan informasi terkait proses bisnis, kebutuhan fitur e-commerce, serta layanan booking workshop yang akan dikembangkan.

2. Desain Sistem

Tahap kedua adalah desain sistem, yaitu merancang struktur sistem yang akan dibangun, meliputi perancangan database, arsitektur backend, serta alur sistem. Perancangan ini bertujuan untuk memberikan gambaran menyeluruh mengenai sistem

sebelum diimplementasikan, termasuk integrasi dengan layanan pembayaran dan pengiriman.

3. Implementasi

Tahap ketiga adalah implementasi, yaitu proses pengembangan sistem berdasarkan desain yang telah dibuat. Pada tahap ini dilakukan pembuatan backend menggunakan framework Laravel, termasuk pembuatan controller, model, routing, serta integrasi dengan database dan layanan pihak ketiga seperti payment gateway dan API pengiriman. Penggunaan Laravel terbukti mampu mendukung pengembangan sistem web yang terstruktur dan efisien.

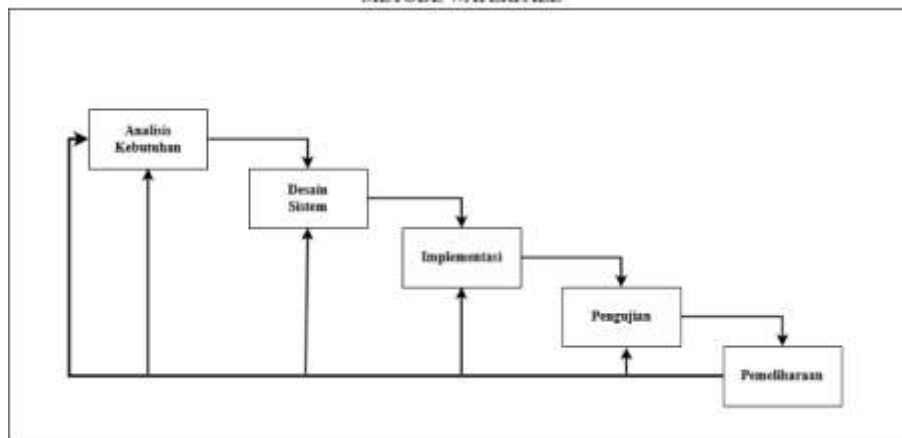
4. Pengujian

Tahap keempat adalah pengujian, yang bertujuan untuk memastikan bahwa sistem yang dikembangkan telah berjalan sesuai dengan kebutuhan. Pengujian dilakukan untuk meminimalisir kesalahan dan memastikan seluruh fitur dapat digunakan dengan baik, termasuk pengujian transaksi pembayaran dan perhitungan ongkos kirim.

5. Pemeliharaan

Tahap terakhir adalah pemeliharaan, yaitu tahap perbaikan dan pengembangan sistem setelah implementasi. Pemeliharaan dilakukan untuk memastikan sistem tetap berjalan optimal serta dapat menyesuaikan dengan kebutuhan pengguna di masa mendatang.

Gambar 1 . Metode Waterfall
METODE WATERFALL



B. Tools dan Teknologi

Dalam penelitian ini, pengembangan sistem dilakukan dengan menggunakan beberapa tools dan teknologi pendukung. Framework utama yang digunakan adalah Laravel, yang berfungsi untuk membangun backend sistem karena memiliki struktur yang rapi, keamanan yang baik, serta mendukung pengembangan berbasis MVC (Model-View-Controller). Database yang digunakan adalah MySQL, yang berfungsi untuk menyimpan dan mengelola data sistem seperti data produk, transaksi, serta data booking workshop.

Selain itu, sistem juga menggabungkan layanan pihak ketiga untuk mendukung fitur e-commerce dengan lebih baik. Payment gateway yang digunakan adalah Tripay, yang berfungsi untuk mengelola transaksi pembayaran secara otomatis melalui berbagai metode pembayaran seperti transfer bank, e-wallet, dan virtual account. Integrasi payment gateway dalam platform e-commerce terbukti mampu meningkatkan keamanan transaksi dan efisiensi proses pembayaran.

Untuk mendukung proses pengiriman barang, sistem menggunakan API RajaOngkir yang berfungsi untuk menghitung ongkos kirim berdasarkan lokasi pengiriman, berat barang, serta jasa ekspedisi yang dipilih. Integrasi API pengiriman seperti RajaOngkir memungkinkan perhitungan ongkir secara otomatis dan real-time sehingga meningkatkan pengalaman pengguna dalam proses checkout.

Tools pendukung yang digunakan dalam proses pengembangan antara lain Visual Studio Code sebagai code editor, Postman untuk melakukan pengujian API, serta browser untuk melakukan pengujian sistem secara langsung.

C. Perancangan Sistem

Perancangan sistem dilakukan untuk memberikan gambaran mengenai alur kerja dan struktur sistem yang akan dikembangkan. Perancangan ini meliputi use case diagram, entity relationship diagram (ERD), serta arsitektur backend. Use case diagram digunakan untuk menggambarkan interaksi antara pengguna dengan sistem, seperti proses registrasi, login, pengelolaan produk, transaksi e-commerce, pemilihan metode pembayaran, serta booking workshop. Entity Relationship Diagram (ERD) digunakan untuk merancang struktur database dan relasi antar tabel yang terdapat dalam sistem, seperti relasi antara pengguna, produk, transaksi, detail transaksi, serta data booking workshop.

1. Usecase Diagram

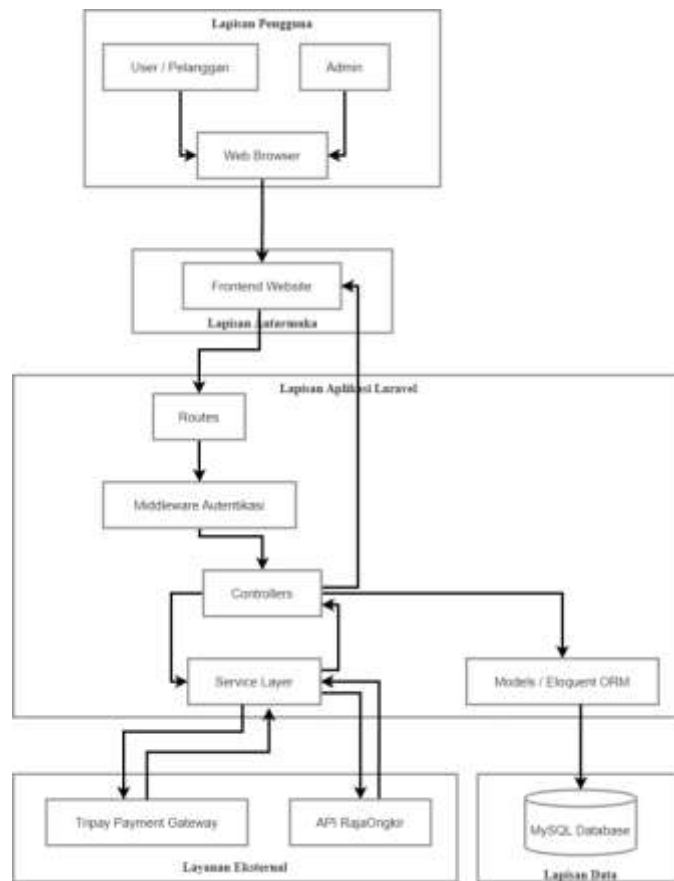
Pada gambar 2 merupakan usecase diagram. Use case diagram menunjukkan peran pengguna dalam sistem. Pengunjung hanya dapat melihat produk dan registrasi. Setelah menjadi pelanggan, pengguna dapat login, menambah atau mengubah keranjang, melakukan checkout, dan memesan workshop. Admin memiliki fungsi manajemen data produk, pesanan, booking, pengguna, serta akses dashboard. Proses checkout terkait dengan perhitungan ongkos kirim dan pembayaran online melalui layanan eksternal.

Gambar 2 . Usecase Diagram



2. Erd Diagram

ERD menggambarkan hubungan antara tabel utama dalam sistem. Produk dikelompokkan berdasarkan kategori, dan setiap produk dapat muncul di keranjang atau pesanan. Pengguna memiliki keranjang, membuat pesanan, dan melakukan booking workshop. Data pembayaran dan pengiriman disimpan secara terpisah tetapi terkait langsung dengan pesanan. Diagram ini memperlihatkan alur data dari pengguna ke produk, order, pembayaran, dan pengiriman, sebagaimana diilustrasikan pada



Selain itu, arsitektur backend dirancang menggunakan konsep MVC yang diterapkan dalam framework Laravel. Arsitektur ini memisahkan antara logika bisnis, pengelolaan data, dan antarmuka, sehingga sistem menjadi lebih terstruktur. Backend juga dirancang untuk mendukung integrasi API eksternal seperti payment gateway dan API pengiriman. Penelitian sebelumnya menunjukkan bahwa integrasi API dalam sistem e-commerce mampu meningkatkan efisiensi transaksi dan akurasi data .

D. Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui beberapa metode. Berikut adalah metode yang digunakan dalam mengumpulkan data dalam penelitian ini:

1. Observasi

Menurut Daruhadi , Observasi adalah teknik pengumpulan data melalui pengamatan langsung terhadap objek atau fenomena yang sedang diteliti. Peneliti melakukan pencatatan aktif terhadap perilaku, kejadian, dan kondisi yang diamati di lapangan tanpa mengintervensi atau mengubah situasi tersebut.

Observasi dilakukan dengan mengamati secara langsung proses operasional UMKM Batik Giri Alam, terutama kegiatan penjualan produk batik dan pendaftaran workshop membuat batik yang masih dilakukan secara manual. Panduan observasi terfokus pada alur pencatatan transaksi, prosedur pemesanan produk, pengelolaan informasi pelanggan, pendaftaran workshop, serta kendala yang timbul dalam layanan pelanggan. Melalui observasi, peneliti dapat mengidentifikasi kendala yang dihadapi, menyesuaikan hasil wawancara dengan kondisi sebenarnya, serta memperoleh data yang lebih akurat sebagai landasan untuk merancang dan mengembangkan sistem backend terintegrasi.

2. Wawancara

Wawancara adalah teknik pengumpulan data yang melibatkan interaksi langsung antara peneliti dan responden, di mana peneliti mengajukan pertanyaan untuk mendapatkan informasi yang mendalam terkait topik yang diteliti. Teknik ini memungkinkan peneliti memahami sudut pandang, pengalaman, maupun opini responden secara rinci .

Wawancara dilakukan secara langsung dengan pihak pengelola UMKM Batik Giri Alam sebagai informan utama untuk memperoleh informasi terkait proses bisnis yang berjalan, kendala dalam pengelolaan penjualan produk batik dan pendaftaran workshop, serta kebutuhan sistem yang diharapkan. Pertanyaan wawancara diarahkan pada alur transaksi, pencatatan data, proses pembayaran, pengelolaan jadwal workshop, serta kebutuhan admin dalam mengelola data penjualan dan reservasi. Metode wawancara ini bertujuan untuk menggali data secara mendalam dan memperoleh gambaran nyata mengenai permasalahan yang akan diselesaikan melalui pengembangan sistem backend.

3. Studi Literatur

Selain itu, studi literatur dilakukan dengan mengkaji berbagai referensi seperti jurnal ilmiah, buku, dan sumber terpercaya lainnya yang berkaitan dengan pengembangan sistem e-commerce, sistem booking online, serta penggunaan framework Laravel. Studi literatur ini bertujuan untuk mendukung dasar teori serta memperkuat penelitian yang dilakukan.

E. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi yang terdapat dalam sistem berjalan dengan baik sesuai dengan kebutuhan. Metode pengujian yang digunakan adalah blackbox testing, yaitu pengujian yang dilakukan dengan menguji fungsi sistem tanpa melihat kode program secara langsung. Pengujian ini berfokus pada input dan output dari sistem untuk memastikan bahwa sistem memberikan hasil yang sesuai. Skenario pengujian disusun berdasarkan kebutuhan fungsional utama sistem. Setiap skenario diuji dengan memberikan input tertentu, menentukan output yang diharapkan, membandingkannya dengan hasil aktual, kemudian memberikan status berhasil atau gagal. Dengan demikian, hasil pengujian tidak hanya menunjukkan bahwa fitur dapat berjalan, tetapi juga memperlihatkan kesesuaian antara kebutuhan pengguna dan fungsi sistem yang dikembangkan.

Selain itu, dilakukan juga pengujian API menggunakan tools seperti Postman untuk memastikan bahwa setiap endpoint API, termasuk integrasi dengan payment gateway dan API RajaOngkir, dapat berjalan dengan baik dan menghasilkan respon yang sesuai. Pengujian terhadap payment gateway dilakukan dengan memastikan proses transaksi, notifikasi pembayaran (callback), serta perubahan status pembayaran berjalan dengan benar. Integrasi payment gateway dalam sistem terbukti mampu meningkatkan efisiensi dan keamanan transaksi digital. Skenario pengujian disusun berdasarkan fitur-fitur utama dalam sistem, seperti pengelolaan produk, proses checkout, transaksi pembayaran, perhitungan ongkir, serta booking workshop. Hasil dari pengujian ini digunakan untuk mengevaluasi kinerja sistem serta memastikan bahwa sistem siap untuk digunakan.

Hasil dan Pembahasan

A. Implementasi Sistem

Pada tahap implementasi, backend Website Batik Giri Alam dibangun menggunakan framework Laravel dengan basis data MySQL. Implementasi sistem mencakup pengembangan modul autentikasi pengguna, manajemen produk, keranjang belanja, proses checkout, integrasi pembayaran melalui Tripay, integrasi biaya pengiriman dari RajaOngkir, pendaftaran workshop, serta dashboard admin. Dalam penelitian ini, fokus perhatian pada implementasi difokuskan pada sisi backend yang bertanggung jawab mengelola logika bisnis, melakukan validasi data, integrasi API eksternal, serta pengolahan transaksi.

Struktur implementasi backend dirancang mengikuti pola MVC (Model, View, Controller) yang disediakan oleh Laravel. Penggunaan Laravel dengan arsitektur MVC mendukung kebutuhan sistem karena memisahkan tanggung jawab antara route, controller, model, dan view. Pemisahan ini menjadikan proses pengelolaan transaksi, pendaftaran workshop, dan administrasi menjadi lebih terstruktur serta mudah untuk dipelihara. Controller berfungsi untuk mengatur alur proses bisnis, model berperan dalam pengelolaan data melalui Eloquent ORM, sedangkan route menjadi penghubung antara permintaan pengguna dan fungsi sistem. Struktur ini sesuai dengan kebutuhan awal Batik Giri Alam yang memerlukan sistem terintegrasi, tetapi tetap mudah dikembangkan ketika terdapat perubahan fitur.

1. Implementasi Routing Sistem

Routing digunakan untuk menghubungkan permintaan pengguna dengan controller atau komponen yang sesuai. Pada sistem ini, route mencakup halaman publik, halaman pelanggan, halaman admin, checkout, pembayaran, booking workshop, serta webhook Tripay.

Potongan kode yang ada di dibawah ini yaitu gambar 5 mengatur routing Laravel dengan cara menghubungkan setiap URL ke controller yang sesuai, sehingga setiap permintaan HTTP diarahkan ke handler yang tepat. Route publik seperti `./shop`, dan `/workshop` tersedia tanpa autentikasi, sedangkan route seperti `/checkout` dan `/workshop/booking` berada di dalam middleware `auth` untuk memastikan hanya pengguna terautentikasi yang dapat mengaksesnya. Route webhook Tripay dibuat tanpa middleware autentikasi karena callback berasal dari server eksternal dan harus diproses tanpa session pengguna.

Gambar 5 . Kode Routing Sistem pada Laravel

```
use Illuminate\Routing\Router;
use Illuminate\Support\Facades\Route;

Route::get('/', [LandingController::class, 'home'])->name('landing.home');
Route::get('/shop', [LandingController::class, 'shop'])->name('landing.shop');
Route::get('/workshop', [WorkshopBookingController::class, 'index'])->name('workshop.index');

Route::middleware(['auth'])->group(function () {
    Route::get('/checkout', [App\Http\Controllers\CheckoutController::class, 'index'])->name('checkout');
    Route::get('/payment/order', [App\Http\Controllers\PaymentController::class, 'show'])->name('payment.checkout');

    Route::post('/workshop/booking', [WorkshopBookingController::class, 'store'])->name('workshop.booking.store');
});

Route::post('/webhooks/tripay', [TripayWebhookController::class, 'handle'])
    ->name('tripay.webhook')
    ->withoutMiddleware(['web']);
```

2. Implementasi Model dan Relasi Data

Model digunakan untuk merepresentasikan struktur data pada sistem. Dalam backend ini, model utama yang digunakan antara lain User, Product, Cart, CartItem, Order, OrderItem, Address, WorkshopBooking, dan WorkshopPayment. Relasi antar model digunakan untuk mendukung proses transaksi e-commerce dan booking workshop, sebagaimana diilustrasikan pada Gambar 6.

Model **Order** seperti pada gambar 6 dibawah ini mendefinisikan atribut yang dapat diisi secara massal melalui **\$fillable**, sehingga field-field penting seperti **user_id**, **order_code**, **status**, **subtotal**, **shipping_cost**, **total**, dan **tripay_reference** dapat disimpan secara otomatis. Properti **\$casts** digunakan untuk mengonversi field JSON seperti **shipping_response** dan **tripay_response** menjadi array PHP, serta **paid_at** menjadi tipe **datetime**, sehingga data eksternal dan waktu pembayaran dapat diproses dengan benar. Relasi **user()** dan **items()** memetakan hubungan many-to-one dengan user dan one-to-many dengan item order, sehingga query data terkait menjadi lebih mudah dan konsisten.

Gambar 6 . Kode Model Order

```
class Order extends Model
{
    protected $fillable = [
        'user_id',
        'order_number',
        'status',
        'payment_status',
        'subtotal',
        'shipping_cost',
        'discount',
        'total',
        'shipping_address',
        'shipping_service',
        'payment_method',
        'tripay_reference',
        'tripay_response',
        'paid_at',
    ];

    protected $casts = [
        'shipping_address' => 'array',
        'tripay_response' => 'array',
        'paid_at' => 'datetime',
    ];

    public function user(): BelongsTo
    {
        return $this->belongsTo(User::class);
    }

    public function items(): HasMany
    {
        return $this->hasMany(WorkshopBooking::class);
    }
}
```

Kode dibawah ini yaitu pada gambar 7 model **WorkshopBooking** merepresentasikan reservasi workshop secara terpisah, dengan atribut seperti **workshop_available_date_id**, **workshop_slot_schedule_id**, **num_participants**, **total_price**, **deposit_amount**, dan **remaining_amount** untuk mencerminkan kebutuhan business logic workshop. Relasi **user()** mengaitkan booking dengan pengguna yang melakukan reservasi, sementara relasi **payments()** memungkinkan satu booking memiliki banyak payment record agar pembayaran bertahap bisa ditangani secara detail. Struktur ini mendukung skema pembayaran deposit dan pelunasan serta memisahkan data booking dari data transaksi biasa.

Gambar 7 . Kode Model Workshop Booking

```
php
class WorkshopBooking extends Model
{
    protected $table = 'workshop_bookings';

    protected $fillable = [
        'user_id',
        'workshop_workshop_id',
        'workshop_workshop_id',
        'booking_number',
        'customer_name',
        'customer_email',
        'customer_phone',
        'new_participants',
        'total_price',
        'deposit_amount',
        'remaining_amount',
        'status',
        'payment_status',
    ];

    public function user(): belongsTo
    {
        return $this->belongsTo(User::class);
    }

    public function payment(): belongsTo
    {
        return $this->belongsTo(Payment::class, 'booking_id');
    }
}
```

B. Implementasi Fitur E-Commerce

1. Implementasi Keranjang dan Checkout

Kode dibawah ini yaitu pada gambar 8 menunjukkan inisialisasi component Livewire untuk proses checkout. Livewire adalah framework real-time Laravel yang memungkinkan update UI tanpa reload halaman. Inisialisasi state ini mendefinisikan tahapan checkout dan variable variabel yang akan di-track secara real-time. Inisialisasi component Livewire untuk checkout menetapkan state awal seperti *currentStep*, *totalSteps*, *selectedAddressId*, *selectedShippingService*, *selectedPaymentMethod*, *shippingCost*, *subtotal*, dan *totalAmount*. Setiap properti ini membantu mengelola alur checkout multi-step, mulai dari pemilihan alamat hingga konfirmasi pembayaran. Dengan state yang tersimpan di component, antarmuka dapat diperbarui secara real-time tanpa perlu reload halaman.

Gambar 8 . Kode Inisialisasi Checkout

```
php
class Checkout extends Component
{
    public $currentStep = 1;
    public $totalSteps = 4;
    public $selectedAddressId = null;
    public $selectedShippingService = null;
    public $selectedPaymentMethod = null;
    public $shippingCost = 0;
    public $subtotal = 0;
    public $totalAmount = 0;
}
```

Kode dibawah ini yaitu pada gambar 9 mendefinisikan method *initializePaymentMethods()* membuat daftar metode pembayaran yang tersedia dengan format terstruktur berupa ID, nama, dan fee setiap metode. Dengan adanya opsi seperti BCA Virtual Account, QRIS, dan DANA, sistem menyediakan fleksibilitas bagi pengguna untuk memilih metode pembayaran yang paling nyaman. Semua metode ini diberi *fee* nol untuk menjaga biaya pembayaran tetap ramah pengguna.

Gambar 9 . Pemilihan Metode Pembayaran pada Checkout

```
php
private function initializePaymentMethods()
{
    $this->paymentMethods = [
        [
            'id' => 'BCAVA',
            'name' => 'BCA Virtual Account',
            'fee' => 0,
        ],
        [
            'id' => 'QRIS',
            'name' => 'QRIS / Scan QR Code',
            'fee' => 0,
        ],
        [
            'id' => 'DANA',
            'name' => 'DANA',
            'fee' => 0,
        ],
    ];
}
```

2. Implementasi Perhitungan Ongkir RajaOngkir

Perhitungan ongkir dilakukan menggunakan service terpisah agar integrasi API lebih modular. Sistem mengirimkan data origin, destination, berat barang, dan kurir ke API RajaOngkir, kemudian hasilnya diformat untuk ditampilkan pada checkout. Integrasi RajaOngkir mendukung kebutuhan sistem dalam menghitung biaya pengiriman secara otomatis berdasarkan alamat tujuan, berat produk, dan ekspedisi yang dipilih. Hal ini memperkuat fungsi e-commerce karena pengguna dapat memperoleh informasi ongkir sebelum menyelesaikan transaksi.

Kode dibawah ini yaitu pada gambar 10 menunjukkan service `getShippingCost()` menghitung biaya pengiriman dengan membangun payload berdasarkan origin, destination, berat, dan daftar kurir yang dipilih, lalu mengirim request ke API RajaOngkir. Apabila respons API berhasil, data biaya diserahkan kembali dalam format yang bisa diproses frontend; jika tidak berhasil, service menggunakan tarif pengiriman default sebagai fallback sehingga proses checkout tetap berjalan. Logika ini memisahkan perhitungan ongkos kirim dari controller dan memastikan sistem lebih tahan terhadap kegagalan layanan eksternal.

Gambar 10 . Kode Service Perhitungan Ongkir RajaOngkir

```
php
public function getShippingCost($originidistrictid, $destinationidistrictid, $weight, $couriers = [])
{
    $payload = [
        'origin' => $originidistrictid,
        'destination' => $destinationidistrictid,
        'weight' => $weight,
    ];

    if (!empty($couriers)) {
        $payload['courier'] = implode('-', $couriers);
    }

    $response = $this->httpClient([
        'key' => $this->apiKey,
    ])->timeout(10)->safeForm()->post($this->calculateCosturl, $payload);

    if ($response->successful()) {
        return $response->json('data') ?? [];
    }

    return $this->getFallbackShippingData();
}
```

3. Implementasi Pembayaran Online Tripay

Pembayaran online dilakukan dengan menghubungkan backend ke gateway Tripay. Saat pengguna mengonfirmasi pesanan, sistem membuat transaksi pembayaran melalui service PaymentService, lalu memperoleh referensi dan URL pembayaran. Integrasi payment gateway Tripay dipilih karena mendukung otomatisasi proses pembayaran dan pembaruan status transaksi. Dengan adanya integrasi ini, admin tidak perlu melakukan pengecekan pembayaran secara manual secara penuh, sehingga proses transaksi menjadi lebih efisien dan risiko kesalahan pencatatan dapat dikurangi.

Kode dibawah ini yaitu pada gambar 11 menunjukkan Class *PaymentService* memisahkan logika pembuatan transaksi pembayaran dari controller dengan menggunakan interface *GatewayInterface* dan implementasi *TripayGateway*. Constructor menginisialisasi gateway Tripay, sedangkan method *createPayment()* mendelegasikan pembuatan transaksi ke gateway tersebut. Desain ini memungkinkan backend untuk tetap fleksibel dan mudah diubah jika nanti ingin mengganti atau menambahkan provider pembayaran lain.

Gambar 11 . Kode Service Pembayaran Tripay

```
php
class PaymentService
{
    protected GatewayInterface $gateway;

    public function __construct()
    {
        $this->gateway = new TripayGateway();
    }

    public function createPayment(array $data): array
    {
        return $this->gateway->createTransaction($data);
    }
}
```

Kode dibawah ini yaitu pada gambar 12 menunjukan method *show()* di controller pembayaran melakukan pemeriksaan otorisasi dengan memastikan order dimiliki oleh user saat ini, lalu menolak akses dengan *abort(403)* jika tidak sesuai. Jika status pembayaran order sudah *paid*, method ini mengalihkan pengguna ke halaman sukses untuk menghindari pembayaran ganda. Jika order belum dibayar, method mengembalikan view checkout yang menampilkan detail order dan opsi pembayaran.

Gambar 12 . Kode Controller Pembayaran Order

```
php
public function show(Order $order)
{
    if ($order->user_id !== Auth::id()) {
        abort(403, 'Unauthorized');
    }

    if ($order->payment_status === 'paid') {
        return redirect()->route('payment.success', $order->id)
            ->with('success', 'Pesanan sudah dibayar');
    }

    return view('checkout.payment', ['order' => $order]);
}
```

C. Implementasi Penyimpanan Booking Workshop

1. Implementasi Pembayaran Booking Workshop

Proses booking workshop dimulai ketika pengguna memilih workshop, jadwal, dan mengisi data peserta. Fitur booking workshop dikembangkan untuk menjawab kebutuhan Batik Giri Alam dalam mengelola reservasi peserta secara lebih tertata. Sistem menyimpan jadwal, jumlah peserta, biaya, deposit, dan status pembayaran sehingga admin dapat memantau reservasi tanpa pencatatan manual. Sistem kemudian memvalidasi jadwal, menghitung biaya total, menentukan deposit, serta menyimpan data booking. Implementasi booking workshop menunjukkan bahwa backend telah dirancang untuk mendukung transaksi reservasi secara lengkap. Data booking tidak hanya menyimpan identitas pengguna, tetapi juga data jadwal, jumlah peserta, total pembayaran, deposit, dan sisa pembayaran.

Pada gambar dibawah ini yaitu gambar 13 yang merupakan potongan kode penyimpanan booking workshop membuat record baru dengan data dari input yang tervalidasi, termasuk *user_id*, *workshop_available_date_id*, *workshop_slot_schedule_id*, *booking_number*, *customer_name*, *customer_email*, *customer_phone*, serta nilai *total_price*, *deposit_amount*, dan *remaining_amount*. Status awal booking diatur menjadi *pending* dan payment status juga pending karena pembayaran belum selesai. Logika ini memastikan seluruh data booking tersimpan secara konsisten dan sesuai dengan aturan bisnis workshop.

Gambar 13 . KodeProses Penyimpanan Booking Workshop

```
php
$booking = workshopBooking::create([
    'user_id' => $user->id,
    'workshop_available_date_id' => $lockedSchedule->available_date_id,
    'workshop_slot_schedule_id' => $validated['workshop_slot_schedule_id'],
    'booking_number' => $bookingNumber,
    'customer_name' => $validated['full_name'],
    'customer_email' => $validated['email'],
    'customer_phone' => $validated['phone'],
    'num_participants' => $validated['number_of_participants'],
    'total_price' => $totalPrice,
    'deposit_amount' => $depositAmount,
    'remaining_amount' => $remainingAmount,
    'status' => 'pending',
    'payment_status' => 'pending',
]);
```

2. Implementasi Fitur Booking Workshop

Pembayaran workshop dilakukan melalui Tripay dengan skema deposit, pelunasan, atau pembayaran penuh. Sistem membuat record pembayaran workshop terlebih dahulu, kemudian mengirim data ke gateway pembayaran. Pada implementasi ini, backend memisahkan data booking dengan data pembayaran workshop.

Kode dibawah ini yaitu pada gambar 14 menunjukkan bagaimana Proses pembayaran workshop dimulai dengan membuat record payment baru melalui relasi *payments()* pada booking, yang menyimpan jumlah bayar, tipe pembayaran, metode, status awal *pending*, dan reference number unik. Setelah record payment dibuat, service pembayaran dipanggil dengan data lengkap seperti *merchant_ref*, *amount*, *method*, dan informasi pelanggan untuk membuat transaksi di Tripay. Pendekatan ini memungkinkan backend melacak setiap transaksi pembayaran dan menyimpan respons Tripay sebagai bagian dari alur pembayaran.

Gambar 14. Kode Proses Pembayaran Booking Workshop

```
php
$payment = $booking->payments()->create([
    'amount' => $amount,
    'type' => $validated['type'],
    'payment_method' => $validated['payment_method'],
    'payment_status' => 'pending',
    'reference_number' => 'TRIPAY_' . Str::random(12),
]);

$response = $paymentService->createPayment([
    'merchant_ref' => $payment->reference_number,
    'amount' => $amount,
    'method' => $validated['payment_method'],
    'customer_name' => $booking->customer_name,
    'customer_email' => $booking->customer_email,
    'customer_phone' => $booking->customer_phone,
]);
```

D. Implementasi Webhook dan Sinkronisasi Status Pembayaran

Webhook adalah mechanism otomatis dimana Tripay mengirim callback ke backend saat user selesai pembayaran. Webhook digunakan agar sistem dapat menerima notifikasi pembayaran dari Tripay secara otomatis. Ketika callback diterima, backend memverifikasi signature, membaca status transaksi, lalu memperbarui status order atau pembayaran workshop.

Kode dibawah ini yaitu pada gambar 15 adalah handler untuk webhook Tripay yang verifies signature dan process payment confirmation. Handler webhook Tripay membaca signature dari header *X-Callback-Signature* dan payload dari body request, lalu membandingkan signature tersebut dengan nilai yang dihitung ulang menggunakan *hash_hmac('sha256', \$payload, \$secretkey)*. Jika signature tidak cocok, request ditolak dengan response 403 untuk menjaga keamanan dan memastikan hanya callback valid yang diproses. Data hasil callback kemudian diambil dari JSON request untuk digunakan dalam pemrosesan lebih lanjut seperti pembaruan status pembayaran.

Gambar 15. Kode Webhook Tripay

```
php
public function handle(Request $request)
{
    $signature = $request->header('X-Callback-Signature');
    $payload = $request->getContent();
    $secretKey = config('services.tripay.private_key');

    $computedSignature = hash_hmac('sha256', $payload, $secretKey);

    if (!hash_equals($computedSignature, $signature)) {
        return response()->json(['error' => 'Invalid signature'], 403);
    }

    $data = $request->json()->all();
    $merchantRef = $data['merchant_ref'] ?? null;
    $reference = $data['reference'] ?? null;
    $status = $data['status'] ?? null;
}
```

Kode dibawah ini yaitu pada gambar 16 menunjukkan bagaimana webhook Tripay di handle untuk update status order. Ini adalah *continuation* dari gambar 15 yang menghandle webhook callback. Logika update status pembayaran mengecek apakah status dari webhook adalah *paid* dan status order saat ini masih *pending*, jika kedua kondisi terpenuhi, method *updateOrderStatus()* dipanggil untuk mengubah status order menjadi *processing*. Dengan mekanisme ini, perubahan status hanya terjadi sekali pada kondisi yang valid, sehingga order yang sudah diproses tidak diubah ulang tanpa alasan. Metode ini juga membantu menjaga konsistensi dan menandai bahwa pembayaran telah dikonfirmasi secara otomatis.

Gambar 16. Kode Update Status Pembayaran Order

```
php
if ($paymentStatus === 'paid' && $order->status === 'pending') {
    $order->updateOrderStatus('processing', 'Pembayaran dikonfirmasi otomatis dari Tripay webhook');
}
```

E. Implementasi Manajemen Admin

Admin memiliki akses untuk mengelola order produk dan booking workshop melalui dashboard backend. Admin dapat memfilter data, melihat detail transaksi, memperbarui status order, dan membatalkan transaksi jika memenuhi syarat. Fitur admin pada backend memungkinkan pengelolaan transaksi secara terpusat. Admin tidak hanya memantau order produk, tetapi juga booking workshop. Implementasi ini penting dalam mendukung kebutuhan operasional Batik Giri Alam agar seluruh transaksi dapat dikelola dalam satu sistem.

Kode dibawah ini yaitu pada gambar 17 menunjukkan Method *updateStatus()* di controller admin memvalidasi input status dan notes untuk memastikan status baru adalah salah satu nilai yang diizinkan dan catatan maksimal 500 karakter. Setelah validasi, order diperbarui dengan method *updateOrderStatus()* yang sama digunakan oleh webhook, sehingga pendekatan perubahan status tetap konsisten. Dengan begitu, admin dapat melakukan override manual pada order sekaligus menjaga keandalan dan auditabilitas proses perubahan status.

Gambar 17. Kode Pengelolaan Order oleh Admin

```
php
public function updateStatus(Request $request, Order $order)
{
    $validated = $request->validate([
        'status' => 'required|in:pending,processing,shipped,delivered,cancelled',
        'notes' => 'nullable|string|max:500',
    ]);

    $order->updateOrderStatus(
        $validated['status'],
        $validated['notes'] ?? null,
        Auth::id()
    );

    return back()->with('success', 'Status pesanan berhasil diubah');
}
```

F. Hasil Pengujian Sistem

Pengujian dilakukan untuk memastikan seluruh fitur backend berjalan sesuai kebutuhan. Berdasarkan hasil implementasi, sistem berhasil mendukung proses registrasi, login, pengelolaan produk, keranjang, checkout, perhitungan ongkir, pembayaran online, booking workshop, dan manajemen admin.

Berdasarkan hasil blackbox testing pada 12 skenario utama, seluruh fitur memperoleh status berhasil sehingga tingkat keberhasilan pengujian mencapai 100%. Hasil ini menunjukkan bahwa fungsi utama sistem telah berjalan sesuai kebutuhan awal pengguna, yaitu mempermudah transaksi produk, menghitung ongkos kirim secara otomatis, memproses pembayaran digital, mengelola booking workshop, serta membantu admin dalam memantau order dan reservasi. Pengujian ini juga menunjukkan bahwa integrasi API eksternal seperti Tripay dan RajaOngkir dapat mendukung proses bisnis secara lebih otomatis dan mengurangi ketergantungan pada pencatatan manual sebagaimana diilustrasikan pada Tabel 1.

Tabel 1. Hasil Pengujian Sistem

No	Fitur	Input	Output yang Diharapkan	Hasil Aktual	Status
1	Registrasi Pengguna	Nama, Email, Password valid	Akun Pengguna berhasil dibuat	Sistem menyimpan data pengguna dan mengarahkan ke halaman login/dashboard	Berhasil
2	Login Pengguna	Email dan password terdaftar	Pengguna Berhasil masuk ke system	System menerima autentikasi dan menampilkan halaman pengguna	Berhasil
3	Tambah produk ke keranjang	Produk dan jumlah pembelian	Produk masuk ke keranjang	Sistem menampilkan order dan menampilkan halaman pembayaran	berhasil
4	Checkout dan pembuatan order	Alamat, ekspedisi, produk, metode pembayaran	Order berhasil dibuat	Sistem menyimpan order dan menampilkan halaman pembayaran	Berhasil
5	Perhitungan ongkir RajaOngkir	Alamat tujuan, berat, produk, kurir	Ongkir tampil sesuai pilihan kurir	Sistem menampilkan opsi ongkir	Berhasil
6	Pembuatan pembayaran Tripay	Order dan metode pembayaran	Transaksi pembayaran dibuat	Sistem menghasilkan instruksi pembayaran	Berhasil
7	Webhook pembayaran order	Callback pembayaran dari Tripay	Status order diperbarui	Status pembayaran/order berubah sesuai callback	Berhasil
8	Booking workshop	Workshop, jadwal, jumlah peserta	Booking berhasil dibuat	Sistem menyimpan data booking	Berhasil
9	Pembayaran deposit workshop	Booking dan metode pembayaran	Pembayaran deposit dibuat	Sistem menghasilkan transaksi deposit	Berhasil
10	Webhook pembayaran workshop	Callback pembayaran workshop	Status pembayaran booking diperbarui	Sistem memperbarui status booking	Berhasil
11	Update status order oleh admin	Status order baru	Status order berubah	Sistem menyimpan perubahan status	Berhasil
12	Admin melihat daftar booking/order	Akses dashboard admin	Data booking/order tampil	Sistem menampilkan daftar transaksi dan booking	Berhasil

Simpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa backend Website Batik Giri Alam berhasil dikembangkan menggunakan framework Laravel dengan metode Waterfall. Sistem yang dibangun mampu mendukung kebutuhan utama layanan digital, yaitu pengelolaan transaksi e-commerce produk batik dan booking workshop dalam satu backend yang terintegrasi. Backend yang dikembangkan telah mencakup fitur registrasi dan login pengguna, pengelolaan produk, keranjang belanja, checkout, perhitungan ongkos kirim menggunakan RajaOngkir, pembayaran online menggunakan Tripay, pengelolaan booking workshop, serta dashboard admin. Implementasi sistem menunjukkan bahwa Laravel mampu memberikan struktur pengembangan yang terorganisasi melalui pola MVC dan memudahkan integrasi dengan layanan pihak ketiga. Berdasarkan hasil blackbox testing terhadap 12 skenario utama, seluruh fitur memperoleh status berhasil sehingga tingkat keberhasilan pengujian mencapai 100%. Hasil tersebut menunjukkan bahwa fungsi transaksi produk, pembayaran online, perhitungan ongkos kirim, booking workshop, webhook pembayaran, serta pengelolaan admin telah berjalan sesuai kebutuhan yang dianalisis sebelumnya. Integrasi payment gateway Tripay dan API RajaOngkir juga mampu mendukung otomatisasi proses pembayaran dan perhitungan ongkir, sehingga dapat mengurangi ketergantungan pada proses manual. Dengan demikian, sistem backend yang dikembangkan dapat membantu Batik Giri Alam dalam meningkatkan efisiensi transaksi produk, mempercepat dan menata proses booking workshop, serta mempermudah admin dalam mengelola data order dan reservasi secara lebih efektif, sistematis, dan terkomputerisasi.

Ucapan Terima Kasih

Penulis mengucapkan puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan karuniaNya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis menyampaikan terima kasih kepada dosen pembimbing atas bimbingan dan arahan yang diberikan, kepada seluruh dosen dan civitas akademika atas ilmu yang telah diberikan, kepada pihak Batik Giri Alam atas dukungan dan data penelitian, serta kepada orang tua, keluarga, dan teman-teman yang senantiasa memberikan doa dan semangat. Penulis menyadari bahwa penelitian ini masih memiliki keterbatasan, sehingga kritik dan saran sangat diharapkan. Semoga penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang sistem informasi dan teknologi web.

References

- [1] B. Putri, N. Amalia, and D. Mairiza, "Eksistensi Prinsip Ekonomi Mikro Islam Terhadap Keberlanjutan Usaha [ISSN 2714-7444 \(online\)](https://doi.org/10.21070/acopen.11.2026.14411), <https://acopen.umsida.ac.id>, published by [Universitas Muhammadiyah Sidoarjo](https://www.muhammadiyah.ac.id)

Mikro Di Era Digitalisasi,” Dec. 2023. doi: <https://doi.org/10.31004/sharing.v2i2.23419>.

[2] S. Qauzilqist and Y. O. Lestari, “Implementasi E-Commerce Terhadap Efisiensi dan Efektivitas UMKM,” vol. 17, no. 3, pp. 211–220, Aug. 2025, doi: [10.24905/permana.v17i3.858](https://doi.org/10.24905/permana.v17i3.858).

[3] I. Islamiati, S. Q. Aini, A. Anisah, and N. Asman, “Peran digitalisasi dalam pemberdayaan ekonomi UMKM melalui aplikasi kasir pintar,” *Jurnal Pembelajaran Pemberdayaan Masyarakat (JP2M)*, vol. 6, no. 2, pp. 545–558, Jun. 2025, doi: [10.33474/jp2m.v6i2.23678](https://doi.org/10.33474/jp2m.v6i2.23678).

[4] S. Asadi and Ferdian, “Penerapan Digitalisasi pada Usaha UMKM Toko Sembako melalui Metode Participatory Action Research (PAR),” Aug. 2025, doi: <https://doi.org/10.55681/swarna.v4i8.1707>.

[5] F. A. Fauzi and F. Darmawan, “Pembangunan Aplikasi E-Commerce berbasis Website Menggunakan Laravel,” Feb. 2023. doi: <https://doi.org/10.23969/pasinformatik.v2i1.7172>.

[6] R. Angel, W. Agustina, Nurhasanah, A. C. Mauluddin, and R. N. Handayani, “Pengembangan Platform E-Commerce UMKM Berbasis Laravel dengan Blackbox Testing dan Metode Waterfall,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 2, Feb. 2025, doi: [10.52436/1.jpti.684](https://doi.org/10.52436/1.jpti.684).

[7] A. Fadhliansyah, E. Wibowo, and K. Panji, “Perancangan Sistem Pemesanan Makanan Berbasis Web Pada Umkm Dengan Framework Laravel,” *Journal of Digital Business and Technology Innovation (DBESTI)*, vol. 2, no. 2, pp. 246–251, Dec. 2025, doi: <https://doi.org/10.54914/dbesti.v2i2.1950>.

[8] Yusni, Supriadi, Revalina, A. Asmarani, Nurfadilla, and D. Tribuana, “Optimalisasi E-Commerce Melalui Teknologi Web Modern (Next.js, Laravel, Dsb): Sebuah Tinjauan Sistematis Literatur (Systematic Literature Review),” *Jurnal Teknologi dan Bisnis Cerdas*, vol. 1, no. 3, pp. 155–170, Dec. 2025, doi: [10.64476/jtbc.v1i3.13](https://doi.org/10.64476/jtbc.v1i3.13).

[9] Irwanto, “Perancangan Sistem Informasi Sekolah Kejuruan dengan Menggunakan Metode Waterfall (Studi Kasus SMK PGRI 1 Kota Serang–Banten),” Feb. 2021. doi: <https://doi.org/10.31849/lectura.v12i1.6093>.

[10] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, “Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API : Kajian Literatur terhadap Framework Laravel dan React,” *Infomatek*, vol. 27, no. 1, pp. 165–178, Jun. 2025, doi: [10.23969/infomatek.v27i1.25122](https://doi.org/10.23969/infomatek.v27i1.25122).

[11] A. R. Chan, A. D. Samala, L. Mursyida, and M. A. Zaus, “Perancangan Website Penjualan dan Implementasi Payment Gateway Toko Arc Fotokopi Berbasis Framework Laravel,” Jun. 2025. doi: <https://doi.org/10.31004/jptam.v9i2.27889>.

[12] W. Andrew, Irwan, W. Susanti, and R. N. Putri, “Aplikasi E-Commerce Menerapkan Payment Gateway Dan Antarmuka Pemograman Aplikasi (Api),” Aug. 2025, doi: <https://doi.org/10.35145/jmapteksi.v7i2.4894>.

[13] A. Handayani and S. Muzid, “Pengembangan Prototype E-commerce Bouquet Bunga Berbasis Laravel dengan Integrasi Rajaongkir dan Midtrans,” vol. 4, no. 5, pp. 1162–1173, Sep. 2025, doi: <https://doi.org/10.53513/jursi.v4i5.11784>.

[14] G. Daruhadi and P. Sopiati, *Pengumpulan Data Penelitian*, vol. 3. 2024. doi: <https://doi.org/10.56799/jceki.v3i5.5181>.

[15] S. Romdona, S. S. Junista, and A. Gunawan, “Teknik Pengumpulan Data: Observasi, Wawancara Dan Kuesioner,” vol. 3, no. 1, pp. 39–47, Jan. 2025, doi: <https://doi.org/10.61787/taceee75>.

[16] D. N. Desiana, K. T. Putri, M. Metravia, and A. Marini, “Studi Pustaka dalam Efektivitas Pemanfaatan Perpustakaan Sekolah untuk Meningkatkan Minat Baca Siswa di Sekolah Dasar,” *Jurnal Pendidikan Guru Sekolah Dasar*, vol. 1, no. 3, p. 15, May 2024, doi: [10.47134/pgsd.v1i3.601](https://doi.org/10.47134/pgsd.v1i3.601).

[17] G. K. Wardana, B. Rahayudi, and W. H. N. Putra, “Pengembangan E-Commerce dengan Integrasi API Payment Gateway Midtrans,” Oct. 2021. Accessed: May 18, 2026. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10097>