TeoriWarna E-Commerce - Complete Documentation

Version: 1.0.0 **Last Updated:** January 4, 2026 **Framework:** Laravel 11 **Admin Panel:** Filament 3

Table of Contents

1. Project Overview

TeoriWarna adalah platform e-commerce modern yang dibangun menggunakan Laravel 11 dengan Filament Admin Panel. Platform ini menyediakan fitur lengkap untuk penjualan produk online dengan sistem pembayaran terintegrasi Midtrans.

Key Features

- ☑ **Product Catalog** - Sistem katalog produk dengan kategorisasi
- ☑ **Shopping Cart** - Keranjang belanja dengan session & database support
- ☑ **Checkout System** - Proses checkout dengan validasi lengkap
- ☑ **Payment Gateway** - Integrasi Midtrans Snap untuk pembayaran
- ☑ **Order Management** - Sistem manajemen pesanan dengan status tracking
- ☑ **User Authentication** - Sistem login/register dengan email verification
- ☑ **Product Reviews** - Sistem review dan rating produk
- ☑ **Admin Panel** - Dashboard admin menggunakan Filament 3
- ☑ **Performance Optimized** - Database indexing & eager loading
- ☑ **Responsive Design** - Mobile-friendly UI dengan Tailwind CSS

Tech Stack

- **Backend:** Laravel 11 (PHP 8.2+)
- **Frontend:** Blade Templates, Alpine.js, Tailwind CSS
- **Admin Panel:** Filament 3
- **Database:** MySQL 8.0+
- **Payment Gateway:** Midtrans Snap
- **Email:** Laravel Mail (Log driver untuk development)

2. System Requirements

Minimum Requirements

- **PHP:** 8.2 or higher
- **Composer:** 2.x
- **MySQL:** 8.0 or higher
- **Node.js:** 18.x or higher
- **NPM:** 9.x or higher

PHP Extensions Required

```
- BCMath
- Ctype
- Fileinfo
- JSON
- Mbstring
- OpenSSL
- PDO
- Tokenizer
- XML
- cURL
```

Recommended Server Configuration

- **Memory Limit:** 256MB minimum
- **Max Execution Time:** 300 seconds
- **Upload Max Filesize:** 20MB
- **Post Max Size:** 25MB

3. Installation Guide

Step 1: Clone & Setup

```
# Clone repository
git clone <repository-url> teoriwarna
cd teoriwarna

# Install PHP dependencies
composer install

# Install Node dependencies
npm install

# Copy environment file
cp .env.example .env
```

```
# Generate application key
php artisan key:generate
```

Step 2: Database Configuration

Edit `.env` file:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=teoriwarna_db
DB_USERNAME=root
DB_PASSWORD=
```

Create database:

```sql
CREATE DATABASE teoriwarna_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

Step 3: Run Migrations & Seeders

```
# Run migrations
php artisan migrate

# Run seeders (creates admin user & sample data)
php artisan db:seed
```

Default Admin Credentials:

- Email: admin@teoriwarna.com
- Password: password

Step 4: Storage & Assets

```
# Create storage symlink
php artisan storage:link

# Build assets
npm run build

# For development with hot reload
npm run dev
```

Step 5: Configure Midtrans

Edit `.env` file:

```
MIDTRANS_SERVER_KEY=SB-Mid-server-YOUR_SERVER_KEY
MIDTRANS_CLIENT_KEY=SB-Mid-client-YOUR_CLIENT_KEY
MIDTRANS_IS_PRODUCTION=false
MIDTRANS_IS_SANITIZED=true
MIDTRANS_IS_3DS=true
```

Get your Midtrans credentials from: https://dashboard.midtrans.com/

Step 6: Optimize Application

```
# Clear all caches
php artisan optimize:clear

# Cache configuration (production only)
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

Step 7: Start Development Server

```
# Start Laravel server
php artisan serve

# Access application
# Customer: http://localhost:8000
# Admin Panel: http://localhost:8000/admin
```

---

4. Architecture & Structure

Directory Structure

```
teoriwarna/
├── app/
│   ├── Filament/        # Filament admin resources
│   │   ├── Resources/   # CRUD resources
│   │   └── Widgets/     # Dashboard widgets
│   ├── Http/
│   │   ├── Controllers/ # Application controllers
│   │   └── Middleware/  # Custom middleware
│   ├── Mail/            # Email templates
│   ├── Models/          # Eloquent models
│   └── Providers/       # Service providers
```

```
├── config/
│   ├── midtrans.php        # Midtrans configuration
│   └── shipping.php        # Shipping costs configuration
├── database/
│   ├── migrations/         # Database migrations
│   └── seeders/            # Database seeders
├── public/
│   ├── css/                # Compiled CSS
│   ├── js/                 # Compiled JavaScript
│   └── storage/            # Public storage (symlink)
├── resources/
│   ├── views/              # Blade templates
│   │   ├── auth/           # Authentication views
│   │   ├── cart/           # Shopping cart
│   │   ├── catalog/        # Product catalog
│   │   ├── checkout/       # Checkout process
│   │   ├── orders/         # Order management
│   │   ├── components/     # Reusable components
│   │   └── layouts/        # Layout templates
│   └── css/                # Source CSS
└── routes/
    ├── web.php             # Web routes
    └── auth.php            # Authentication routes
```

MVC Architecture

Models (app/Models/):

- `User.php` - User authentication & profiles
- `Product.php` - Product catalog
- `Category.php` - Product categories
- `Cart.php` - Shopping cart
- `CartItem.php` - Cart items
- `Order.php` - Customer orders
- `OrderItem.php` - Order line items
- `Payment.php` - Payment records
- `ProductReview.php` - Product reviews
- `UserAddress.php` - Shipping addresses

Controllers (app/Http/Controllers/):

- `ProductCatalogController.php` - Product browsing
- `CartController.php` - Cart management
- `CheckoutController.php` - Checkout process
- `OrderController.php` - Order management
- `PaymentController.php` - Payment processing
- `ReviewController.php` - Review management
- `AddressController.php` - Address management

Views (resources/views/):

- Blade templating engine
- Component-based architecture
- Responsive design with Tailwind CSS
- Alpine.js for interactivity

---

5. Features Documentation

5.1 Product Catalog

Features:

- Product listing with pagination
- Category filtering
- Search functionality
- Price range filtering
- Sort options (latest, price, name)
- Product detail pages
- Image gallery
- Related products

Routes:

```
GET /catalog - Product listing
GET /catalog/{product} - Product detail
```

Key Files:

- app/Http/Controllers/ProductCatalogController.php
- resources/views/catalog/index.blade.php
- resources/views/catalog/show.blade.php

5.2 Shopping Cart

Features:

- Add to cart
- Update quantity
- Remove items
- Session-based cart for guests
- Database cart for authenticated users
- Auto-sync on login
- Real-time cart count in navigation

Routes:

```
GET /cart - View cart
POST /cart/add - Add item
```

```
POST /cart/update/{cartItem} - Update quantity
POST /cart/remove/{cartItem} - Remove item
GET /cart/count - Get cart count (AJAX)
```

Key Files:

- app/Http/Controllers/CartController.php
- resources/views/cart/index.blade.php
- app/Models/Cart.php
- app/Models/CartItem.php

5.3 Checkout System

Features:

- Multi-step checkout
- Shipping information form
- Saved addresses
- Shipping method selection (Regular, Express, Same Day)
- Payment method selection (Bank Transfer, E-Wallet, COD)
- Order notes
- Stock validation
- Price consistency check

Routes:

```
GET /checkout - Checkout page
POST /checkout/process - Process checkout
POST /buy-now/{product} - Buy now
GET /checkout/buy-now - Buy now checkout
POST /checkout/buy-now/process - Process buy now
```

Shipping Costs: Configured in config/shipping.php:

- Regular (3-5 days): Rp 15,000
- Express (1-2 days): Rp 30,000
- Same Day: Rp 50,000

Key Files:

- app/Http/Controllers/CheckoutController.php
- resources/views/checkout/index.blade.php
- resources/views/checkout/buy-now.blade.php
- config/shipping.php

5.4 Payment Integration (Midtrans)

Features:

- Midtrans Snap payment popup
- Multiple payment methods (Bank Transfer, E-Wallet, Credit Card, etc.)
- Payment webhook for status updates
- Transaction tracking
- Auto order status update

Payment Flow:

1. User creates order → Order status: `pending`, Payment status: `pending`
2. User clicks "Bayar Sekarang" → Generate Snap token
3. Midtrans popup opens → User completes payment
4. Midtrans sends webhook → Update payment & order status
5. Order status updated to `processing` or `failed`

Routes:

```
POST /payment/{order}/snap-token - Generate payment token
POST /payment/notification - Midtrans webhook (no auth)
GET /payment/{order}/finish - Payment finish page
GET /payment/{order}/status - Check payment status
```

Payment Statuses:

- `pending` - Awaiting payment
- `paid` - Payment successful
- `failed` - Payment failed
- `cancelled` - Payment cancelled

Key Files:

- `app/Http/Controllers/PaymentController.php`
- `config/midtrans.php`
- `app/Models/Payment.php`

Midtrans Configuration:

```php
// config/midtrans.php
return [
    'server_key' => env('MIDTRANS_SERVER_KEY'),
    'client_key' => env('MIDTRANS_CLIENT_KEY'),
    'is_production' => env('MIDTRANS_IS_PRODUCTION', false),
    'is_sanitized' => env('MIDTRANS_IS_SANITIZED', true),
    'is_3ds' => env('MIDTRANS_IS_3DS', true),
];
```

5.5 Order Management

Features:

- Order listing
- Order detail view
- Order status tracking
- Cancel order (pending only)
- Complete order (shipped status)
- Email notifications

Order Statuses:

- `pending` - Awaiting payment
- `processing` - Payment received, processing order
- `shipped` - Order shipped
- `delivered` - Order delivered
- `cancelled` - Order cancelled

Routes:

```
GET /orders - Order list
GET /orders/{order} - Order detail
POST /orders/{order}/cancel - Cancel order
POST /orders/{order}/complete - Mark as received
```

Email Notifications:

- Order confirmation (to customer)
- New order notification (to admin)

Key Files:

- `app/Http/Controllers/OrderController.php`
- `resources/views/orders/index.blade.php`
- `resources/views/orders/show.blade.php`
- `app/Mail/OrderConfirmation.php`
- `app/Mail/AdminNewOrder.php`

5.6 Product Reviews

Features:

- Star rating (1-5)
- Written review
- Review moderation
- Review listing on product page
- Edit/delete own reviews

Routes:

```
GET /reviews - My reviews
POST /catalog/{product}/reviews - Create review
GET /reviews/{review}/edit - Edit form
PUT /reviews/{review} - Update review
DELETE /reviews/{review} - Delete review
```

Key Files:

- app/Http/Controllers/ReviewController.php
- app/Models/ProductReview.php

5.7 User Authentication

Features:

- Registration
- Login/Logout
- Email verification
- Password reset
- Profile management

Routes:

```
GET /register - Registration form
POST /register - Register user
GET /login - Login form
POST /login - Authenticate
POST /logout - Logout
GET /verify-email - Email verification
POST /forgot-password - Password reset request
```

Default Users (from seeder):

Admin:

- Email: admin@teoriwarna.com
- Password: password

Regular User:

- Email: user@example.com
- Password: password

5.8 Admin Panel (Filament)

Access: http://localhost:8000/admin

Features:

- Dashboard with statistics
- Product management (CRUD)
- Category management
- Order management
- Payment tracking
- User management
- Review moderation
- Sales analytics

Admin Resources:

- `app/Filament/Resources/ProductResource.php`
- `app/Filament/Resources/CategoryResource.php`
- `app/Filament/Resources/OrderResource.php`
- `app/Filament/Resources/PaymentResource.php`
- `app/Filament/Resources/UserResource.php`
- `app/Filament/Resources/ReviewResource.php`

Widgets:

- `app/Filament/Widgets/SalesChart.php` - Sales overview

---

6. Database Schema

Users Table

```
users
├── id (PK)
├── name
├── email (unique)
├── password
├── phone
├── role (enum: admin, user)
├── is_active (boolean)
├── is_banned (boolean)
├── profile_picture
├── last_login
├── email_verified_at
└── timestamps
```

Products Table

```
products
├── id (PK)
├── category_id (FK)
├── name
├── slug (unique)
```

```
├── description
├── price (decimal)
├── stock (integer)
├── status (enum: active, inactive, archived)
└── timestamps

Indexes:
- category_id
- status
- (status, category_id) - composite
```

Categories Table

```
categories
├── id (PK)
├── name
├── slug (unique)
├── description
├── is_active (boolean)
└── timestamps
```

Product Images Table

```
product_images
├── id (PK)
├── product_id (FK)
├── image_path
├── is_primary (boolean)
└── timestamps

Indexes:
- product_id
- (product_id, is_primary) - composite
```

Carts Table

```
carts
├── id (PK)
├── user_id (FK)
└── timestamps
```

Cart Items Table

```
cart_items
├── id (PK)
├── cart_id (FK)
├── product_id (FK)
├── quantity (integer)
├── price (decimal)
└── timestamps

Indexes:
- cart_id
- product_id
```

Orders Table

```
orders
├── id (PK)
├── user_id (FK)
├── address_id (FK, nullable)
├── order_number (unique)
├── total_amount (decimal)
├── subtotal (decimal)
├── shipping_cost (decimal)
├── shipping_method (enum)
├── shipping_name
├── shipping_phone
├── shipping_address
├── shipping_city
├── shipping_postal_code
├── tracking_number
├── shipping_courier
├── notes
├── status (enum)
└── timestamps

Indexes:
- user_id
- status
```

Order Items Table

```
order_items
├── id (PK)
├── order_id (FK)
├── product_id (FK)
├── quantity (integer)
├── price (decimal)
├── subtotal (decimal)
└── timestamps
```

```
Indexes:
- order_id
- product_id
```

Payments Table

```
payments
├── id (PK)
├── order_id (FK)
├── payment_method (varchar)
├── method (varchar)
├── amount (decimal)
├── status (enum)
├── snap_token (Midtrans)
├── transaction_id (Midtrans)
├── payment_date
└── timestamps
```

Product Reviews Table

```
product_reviews
├── id (PK)
├── product_id (FK)
├── user_id (FK)
├── rating (1-5)
├── review_text
├── status (enum: pending, approved, rejected)
└── timestamps

Indexes:
- product_id
- user_id
```

User Addresses Table

```
user_addresses
├── id (PK)
├── user_id (FK)
├── label (e.g., "Home", "Office")
├── recipient_name
├── phone
├── address
├── city
├── postal_code
```

```
├── is_default (boolean)
└── timestamps
```

---

7. API Integration

Midtrans Snap API

Base URL:

- Sandbox: https://app.sandbox.midtrans.com
- Production: https://app.midtrans.com

Authentication: Uses Server Key (Base64 encoded)

Snap Token Generation:

```php
$params = [
    'transaction_details' => [
        'order_id' => $order->order_number,
        'gross_amount' => (int) $order->total_amount,
    ],
    'customer_details' => [
        'first_name' => $order->user->name,
        'email' => $order->user->email,
        'phone' => $order->shipping_phone,
    ],
    'item_details' => [
        // Order items
    ],
];

$snapToken = \Midtrans\Snap::getSnapToken($params);
```

Webhook Notification:

```php
// POST /payment/notification
$notification = new \Midtrans\Notification();

$transactionStatus = $notification->transaction_status;
$fraudStatus = $notification->fraud_status;

if ($transactionStatus == 'capture') {
    if ($fraudStatus == 'accept') {
        // Payment success
    }
} else if ($transactionStatus == 'settlement') {
    // Payment success
} else if ($transactionStatus == 'pending') {
```

```
        // Payment pending
    } else if ($transactionStatus == 'deny' ||
            $transactionStatus == 'expire' ||
            $transactionStatus == 'cancel') {
        // Payment failed/cancelled
    }
```

8. User Guide

8.1 Creating an Account

1. Click "Register" in navigation
2. Fill in registration form:
    - Name
    - Email
    - Phone (optional)
    - Password (min 8 characters)
3. Click "Register"
4. Verify email (check inbox)
5. Login with credentials

8.2 Browsing Products

1. Go to "Belanja" menu
2. Use filters:
    - Search by name
    - Filter by category
    - Set price range
    - Sort by price/name/latest
3. Click product for details
4. View product images, description, reviews

8.3 Adding to Cart

Option 1: Add to Cart

1. On product page, select quantity
2. Click "Add to Cart"
3. Continue shopping or view cart

Option 2: Buy Now

1. On product page, select quantity
2. Click "Buy Now"
3. Goes directly to checkout

8.4 Checkout Process

1. Go to cart, select items
2. Click "Checkout Selected Items"
3. Fill shipping information:
   - Name, phone, address
   - City, postal code
4. Select shipping method:
   - Regular: Rp 15,000 (3-5 days)
   - Express: Rp 30,000 (1-2 days)
   - Same Day: Rp 50,000
5. Select payment method:
   - Bank Transfer
   - E-Wallet (OVO, GoPay, DANA)
   - COD
6. Add notes (optional)
7. Click "Place Order"

8.5 Payment Process

1. Order created, redirected to order detail
2. Click "▭ Bayar Sekarang"
3. Midtrans Snap popup opens
4. Choose payment method:
   - Credit/Debit Card
   - Bank Transfer (BCA, Mandiri, BNI, etc.)
   - E-Wallet (GoPay, ShopeePay, QRIS)
   - Convenience Store (Alfamart, Indomaret)
5. Complete payment
6. Order status automatically updated

8.6 Tracking Orders

1. Go to "Pesanan Saya" menu
2. View order list with status
3. Click order for details
4. Check order status:
   - Pending: Awaiting payment
   - Processing: Payment received
   - Shipped: On delivery
   - Delivered: Completed
5. If shipped, view tracking number
6. Click "Barang Sudah Diterima" when received

8.7 Writing Reviews

1. Go to completed order
2. Click product name
3. Scroll to reviews section
4. Click "Write a Review"

5. Select rating (1-5 stars)

6. Write review text

7. Submit review

---

9. Admin Guide

9.1 Accessing Admin Panel

1. Go to `http://localhost:8000/admin`

2. Login with admin credentials

3. Access dashboard

9.2 Managing Products

Create Product:

1. Go to "Products" menu

2. Click "New Product"

3. Fill product details:
   - Name, slug
   - Category
   - Description
   - Price, stock
   - Status (active/inactive)

4. Upload images (set primary)

5. Click "Create"

Edit Product:

1. Click product from list

2. Edit details

3. Click "Save"

Delete Product:

1. Click product

2. Click "Delete"

3. Confirm

9.3 Managing Orders

View Orders:

1. Go to "Orders" menu

2. Filter by status

3. Click order for details

Update Order Status:

1. Open order

2. Change status dropdown:
   - Pending → Processing (after payment)
   - Processing → Shipped
   - Shipped → Delivered
3. Add tracking number if shipped
4. Click "Save"

Cancel Order:

1. Open order (pending only)
2. Click "Cancel Order"
3. Stock automatically restored

9.4 Managing Categories

1. Go to "Categories" menu
2. Create/Edit/Delete categories
3. Toggle active status

9.5 Managing Users

1. Go to "Users" menu
2. View user list
3. Ban/unban users
4. Change user roles

9.6 Viewing Analytics

1. Dashboard shows:
   - Total revenue
   - Orders count
   - Products count
   - Users count
2. Sales chart (monthly)
3. Recent orders

---

10. Configuration

10.1 Environment Variables

```
# App
APP_NAME=Laravel
APP_ENV=local
APP_DEBUG=false
APP_URL=http://localhost

# Database
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
DB_DATABASE=teoriwarna_db
DB_USERNAME=root
DB_PASSWORD=

# Cache
CACHE_STORE=file
CACHE_PREFIX=teoriwarna

# Mail
MAIL_MAILER=log
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

# Midtrans
MIDTRANS_SERVER_KEY=SB-Mid-server-YOUR_SERVER_KEY
MIDTRANS_CLIENT_KEY=SB-Mid-client-YOUR_CLIENT_KEY
MIDTRANS_IS_PRODUCTION=false
MIDTRANS_IS_SANITIZED=true
MIDTRANS_IS_3DS=true
```

10.2 Shipping Configuration

Edit `config/shipping.php`:

```php
return [
    'costs' => [
        'regular' => 15000,   // 3-5 days
        'express' => 30000,   // 1-2 days
        'same_day' => 50000,  // Same day
    ],
    'default_method' => 'regular',
];
```

10.3 Filament Configuration

Edit `app/Providers/Filament/AdminPanelProvider.php`:

```php
public function panel(Panel $panel): Panel
{
    return $panel
        ->default()
        ->id('admin')
        ->path('admin')
        ->login()
        ->colors([
            'primary' => Color::Amber,
        ])
        ->discoverResources(in: app_path('Filament/Resources'), for:
```

```
    'App\\Filament\\Resources')
            ->discoverPages(in: app_path('Filament/Pages'), for:
    'App\\Filament\\Pages')
            ->discoverWidgets(in: app_path('Filament/Widgets'), for:
    'App\\Filament\\Widgets')
            ->widgets([
                Widgets\AccountWidget::class,
                Widgets\FilamentInfoWidget::class,
            ])
            ->middleware([
                EncryptCookies::class,
                AddQueuedCookiesToResponse::class,
                StartSession::class,
                AuthenticateSession::class,
                ShareErrorsFromSession::class,
                VerifyCsrfToken::class,
                SubstituteBindings::class,
                DisableBladeIconComponents::class,
                DispatchServingFilamentEvent::class,
            ])
            ->authMiddleware([
                Authenticate::class,
            ]);
    }
```

11. Troubleshooting

Common Issues

1. "Class not found" Error

Solution:

```
composer dump-autoload
php artisan optimize:clear
```

2. "SQLSTATE[HY000] [2002] Connection refused"

Solution:

- Check MySQL is running
- Verify DB credentials in `.env`
- Test connection: `php artisan migrate:status`

3. "The GET method is not supported"

Solution:

```
php artisan route:clear
php artisan optimize:clear
```

4. Images Not Displaying

Solution:

```
php artisan storage:link
```

Check `public/storage` symlink exists.

5. Midtrans Payment Not Working

Solution:

- Verify Midtrans keys in `.env`
- Check sandbox/production mode
- Check webhook URL is accessible
- View Laravel logs: `storage/logs/laravel.log`

6. Cart Count Not Updating

Solution:

```
php artisan view:clear
php artisan optimize:clear
```

Clear browser cache.

7. "Class 'Midtrans\Config' not found"

Solution:

```
composer require midtrans/midtrans-php
composer dump-autoload
```

8. Performance Issues

Solution:

```
# Enable caching (production only)
php artisan config:cache
```

```
php artisan route:cache
php artisan view:cache

# Run migrations for indexes
php artisan migrate

# Set APP_DEBUG=false in .env
```

Database Indexes

Performance indexes are automatically created by migration:

database/migrations/2026_01_04_044225_add_performance_indexes_to_tables.php

Indexes on:

- products.category_id
- products.status
- cart_items.cart_id
- cart_items.product_id
- order_items.order_id
- product_images.product_id
- orders.user_id
- orders.status

Logs Location

- Laravel Logs: storage/logs/laravel.log
- Filament Logs: storage/logs/filament.log
- Web Server Logs: Check Apache/Nginx logs

Debug Mode

For development, enable debug mode in .env:

```
APP_DEBUG=true
APP_ENV=local
```

⚠ **WARNING:** Never enable APP_DEBUG=true in production!

Appendix A: Artisan Commands

```
# Cache commands
php artisan cache:clear          # Clear application cache
php artisan config:clear         # Clear config cache
php artisan route:clear          # Clear route cache
php artisan view:clear           # Clear view cache
php artisan optimize:clear       # Clear all caches
```

```
# Cache generation (production)
php artisan config:cache        # Cache configuration
php artisan route:cache         # Cache routes
php artisan view:cache          # Cache views

# Database
php artisan migrate             # Run migrations
php artisan migrate:fresh --seed  # Fresh DB with seeders
php artisan db:seed             # Run seeders only

# Storage
php artisan storage:link        # Create storage symlink

# Filament
php artisan filament:user       # Create Filament user

# Queue
php artisan queue:work          # Process queue jobs
php artisan queue:listen        # Listen for jobs
```

Appendix B: Testing Credentials

Midtrans Sandbox

Test Credit Cards:

- Card Number: 4811 1111 1111 1114
- CVV: 123
- Exp: Any future date
- OTP: 112233

Test VA Numbers:

- BCA: Auto-success after generation
- Mandiri: Auto-success after generation

Test E-Wallet:

- GoPay: Use QR code or deep link
- ShopeePay: Auto-success

Email Testing

Development uses log driver. Check emails in: storage/logs/laravel.log

For real emails, configure SMTP:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
```

```
MAIL_PORT=2525
MAIL_USERNAME=your-username
MAIL_PASSWORD=your-password
```

Appendix C: Performance Optimization Checklist

- ☑ Database indexes added
- ☑ Eager loading implemented
- ☑ N+1 query problems resolved
- ☑ APP_DEBUG=false
- ☑ Cache configuration
- ☑ Optimized image loading
- ☑ Consistent pricing (no cached prices)
- ☑ Shipping costs in config file
- ☐ Production server optimization:
    - ☐ OPcache enabled
    - ☐ Redis/Memcached for cache
    - ☐ Queue workers running
    - ☐ CDN for static assets
    - ☐ Database query optimization

Appendix D: Security Checklist

- ☑ CSRF protection enabled
- ☑ SQL injection prevention (Eloquent ORM)
- ☑ XSS protection (Blade escaping)
- ☑ Password hashing (bcrypt)
- ☑ Email verification
- ☑ HTTPS ready
- ☐ Production security:
    - ☐ Rate limiting configured
    - ☐ SSL certificate installed
    - ☐ Secure headers configured
    - ☐ Backup strategy implemented
    - ☐ Regular security updates

Support & Contact

For issues or questions:

- GitHub Issues: `<repository-url>/issues`
- Email: admin@teoriwarna.com
- Documentation: This file

Generated with Claude Code