

DAFTAR PUSTAKA

- Abdi, R. (2018). Rancang Bangun Sistem Informasi *E-commerce* Penjualan Buku Berbasis Web (Studi KAsus : PT. Agromedia Group). In *SKRIPSI*. Pekanbaru: Universitas Islam Negeri Sultan Syarif Kasim Riau.
- Abraham, J., & Ismail, I. E. (2021). Unit Testing dan User Acceptance Testing pada Sistem Informasi Pelayanan Kategorial Pelayanan Anak.
- Armanda, T., & Putra, A. D. (2020). Rancang Bangun Aplikasi *E-commerce* untuk Penjualan Helm. *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, 1, No.1, 17-24.
- Asmarajaya, K. A., Sanjaya, K. O., Putra, D. M., Mahendra, G. S., & Hasanah, F. N. (2021). Sistem Informasi Keuangan Pada Perusahaan Kost Elit Dengan Metode Waterfall. *Jurnal Swabumi*, 107-116. Retrieved from <https://ejournal.bsi.ac.id/ejurnal/index.php/swabumi/article/view/10970/pdf>
- Dedi, Sutarman, & Septiyani, N. (2020). Sistem Informasi *E-commerce* Berbasis Web Pada Toko Indonesia Okubo Jepang. *AJCSR [Academic Journal of Computer Science Research]*.
- Hastanti, R. P., Purnama, B. E., & Wardati, I. U. (2015). Sistem Penjualan Berbasis Web (*E-commerce*) Pada Tata Distro Kabupaten Pacitan. *Jurnal Bianglala Informatika Vol.3*, 1-9.
- Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Jogjakarta: Andi.
- Kosasi, S. (2015). Perancangan Sistem *E-commerce* Untuk Memperluas Pasar Produk Oleh-Oleh Khas Pontianak. *Seminar Nasional Teknologi Informasi dan Komunikasi 2015 (SENTIKA 2015)*, 110-119.
- Kurniawan, B., & Romzi, M. (2021). Perancangan Sistem Pemesanan Makanan Berbasis Web Menggunakan PHP dan MySQL. *Jurnal Informatika dan Komputer*, 12, 2089-4384.

- Mulia, G. S., Najoan, X. B., & Lumenta, A. S. (2022). Analisa Teknologi Hyper Text Markup Language (HTML) VERSI 5. *Jurnal Teknik Informatika*, Vol.15.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Jurnal Ilmiah Teknologi Informasi Terapan. *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)*, 31-36.
- Oktavianti, G. (2019, Maret). Pengantar Sistem Informasi. *Pengantar Sistem Informasi*, 2, 31. Retrieved from <https://www.researchgate.net/publication/331672535>
- Perry, W. E. (2006). *Effective Methods for Software Testing, Third Edition*. Indiana: Wiley.
- Prilantana, K. L., Putra, N. T., & Thalib, E. F. (2021). Rancang Bangun Sistem Informasi *E-commerce* Berbasis Web Pada Concordia Music Shop. *Jurnal Penelitian Saintek*, 26, 178-188.
- Ramdiasyah, Ramdiansyah, F. D., & Anubakti, D. (2020). Perancangan *E-commerce* Pada Bintang Fashion Menggunakan PHP & Laravel Framework. *Jurnal IDEALIS Vol.3 No.1*, 457-461.
- Rudy. (2010). Perancangan Model Bisnis dan Antarmuka untuk Penjualan Online (Studi Kasus: Penjualan Perangkat Komputer). *Seminar Nasional Aplikasi Teknologi Informasi*, 149-152.
- Setiawan, A. A., Lumenta, A. S., & Sompie, S. R. (2019). Rancang Bangun Aplikasi Unsrat E-Catalogue. *Jurnal Teknik Informatika*, Vol. 14, 1-9.
- Setyawan, R., & Maryam. (2021). Sistem Informasi Penjualan Alat Elektronik Berbasis Web Pada Toko Mandiri Elektronik Purwantoro. *Jurnal Digital Teknologi Informasi Vol.8 No.1*, 8-17.
- Sidik, B. (2014). *Pembrograman Web dengan PHP*. Bandung: Informatika.
- Simarmata, J. (2010). *Rekayasa Web*. Penerbit Andi.

- Supardi, J., & Dores, V. (2009, Agustus). Rancang Bangun Collaborative System Pemasaran Hotel Secara Online dengan Pendekatan Mediator Based. *Jurnal Sistem Informasi (JSI)*, VOL. 1, NO. 2, pp55-61.
- Supardi, Y. (2010). *Analisis dan Desain Sistem Informasi*. Jakarta: PT Gramedia Pustaka Utama.
- Susilo, M., Kurniati, R., & Kasmawati. (2018, Maret). Rancang Bangun Website Toko Online Menggunakan Metode Waterfall. *InfoTekJar*, Vol.2, 98-105.
- Sutarman, D., & Septyani, N. (2020). Sistem Informasi *E-commerce* Berbasis Web Pada Toko Indonesia Okubo Jepang. *AJCSR*.
- Wahyunningrum, T., & Januarita, D. (2015). Implementasi dan Pengujian Web *E-commerce* untuk Produk Unggulan Desa. *Jurnal Komputer Terapan*, Vol.1, No.1, 55-66.
- Yang, Y., Ke, W., Yang, J., & Li, X. (2019). Integrating UML With Service Refinement for Requirements Modeling and Analysis. *IEEE Access*, Vol.7, 11599-11612. doi:10.1109/ACCESS.2019.2892082

LAMPIRAN

Lampiran 1

app/Http/Controllers/Auth/LoginController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Models\User;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Str;

class LoginController extends Controller
{
    public function index()
    {
        return view('auth.login');
    }

    public function indexAdmin()
    {
        return view('auth.login-admin');
    }

    public function authenticate(Request $request)
    {
        $credentials = $request->validate([
            'email' => ['required', 'email'],
            'password' => ['required'],
        ]);

        if (Auth::attempt($credentials)) {
            $request->session()->regenerate();
            $username = Auth::user()->name;

            return redirect('/')->with('loginUser',
$username);
        }
    }
}
```

```

        return back()->with('loginError', 'Email atau
password tidak sesuai!');
    }

    public function authenticateAdmin(Request $request)
    {
        $credentials = $request->validate([
            'email' => ['required', 'email'],
            'password' => ['required'],
        ]);

        if (Auth::attempt($credentials)) {
            $request->session()->regenerate();

            return redirect('/dashboard');
        }

        return back()->with('loginError', 'Email atau
password tidak sesuai!');
    }

    public function logout()
    {
        Auth::logout();
        return redirect('/');
    }

    public function forgotPassword()
    {
        return view('auth.forgot');
    }

    public function createToken(Request $request)
    {
        $request->validate([
            'email' => 'required|email|exists:users',
        ]);

        $token = Str::random(64);

        $user = DB::table('password_reset_tokens')-
        >where('email', $request->email)->count();

        if ($user != 0) {
            DB::table('password_reset_tokens')
                ->where('email', $request->email)
                ->update(['token' => $token]);
        }
    }

```

```

    } else {
        DB::table('password_reset_tokens')->insert([
            'email' => $request->email,
            'token' => $token,
            'created_at' => Carbon::now()
        ]);
    }

    Mail::send('auth.forgot_password_message', ['token'
=> $token], function ($message) use ($request) {
        $message->to($request->email);
        $message->subject('Reset Password');
    });

    return back()->with('success', 'Silahkan Cek Email
Anda!');
}

public function showResetPasswordForm($token)
{
    return view('auth.forgot_create_password', [
        'token' => $token,
        "title" => "Forget Password"
    ]);
}

public function createNewPassword(Request $request)
{
    $request->validate([
        'email' => 'required|email|exists:users',
        'password' => 'required|confirmed',
    ], [
        'exists' => 'Masukkan Email Yang Anda Gunakan
Mendaftar!',
        'confirmed' => 'Password Berbeda Dengan
Password Confirmation!'
    ]);

    $updatePassword =
DB::table('password_reset_tokens')
        ->where([
            'email' => $request->email,
            'token' => $request->token
        ])->first();

    if (!$updatePassword) {

```

```

        return back()->with('failed', 'Masukkan Email
Yang Anda Gunakan Mendaftar!');
    }

    $user = User::where('email', $request->email)-
>update(['password' => Hash::make($request->password)]);

    DB::table('password_reset_tokens')->where(['email'
=> $request->email])->delete();

    return redirect('/login')->with('success',
'Password Anda Berhasil Diubah!');
    }
}

```

app/Http/Controllers/User/OrderController.php

```

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;
use App\Models\Order;
use App\Models\User;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\View\View;

class OrderController extends Controller
{
    public function orders(): View
    {
        $this->cancelExpiredOrder();
        $user = User::find(auth()->user()->id);
        $orders = $user->orders()->orderBy('updated_at',
'desc')->with('orderDetails')->get();

        $orders = $orders->map(function ($order) {
            $order->total_qty = $order->orderDetails-
>sum('qty');
            $order->first_image = $order->orderDetails-
>first()->product->images->first()->image_url ?? null;
            return $order;
        });
    }
}

```

```

        return view('user.account_orders',
compact('orders'));
    }

    public function orderDetail($orderNumber)
    {
        $user = User::find(auth()->user()->id);
        $order = $user->orders()->where('order_number',
$orderNumber)->with('orderDetails')->firstOrFail();
        return view('user.order_detail', compact('order'));
    }

    public function cancelOrder(Request $request)
    {
        $this->cancelOrderByid($request['order_number']);

        return redirect()->route('user.orders');
    }

    public function confirmPayment(Request $request)
    {
        $validated = $request->validate([
            'order_number' => ['required'],
            'paid_at' => ['required'],
        ]);

        if ($request->hasFile('payment_proof')) {
            $filename = uniqid() . '.' . $request-
>file('payment_proof')->getClientOriginalExtension();
            $request->file('payment_proof')-
>storeAs('public/payment_proof', $filename);
            $validated['payment_proof'] = $filename;
        } else {
            return back();
        }

        Order::where('order_number',
$validated['order_number'])->update([
            'paid_at' => $validated['paid_at'],
            'payment_proof' => $validated['payment_proof'],
            'payment_status' => 1,
        ]);

        return redirect()->route('user.orders')-
>with('success', 'Silahkan melihat status pesanan Anda di
halaman pesanan');
    }

```

```

public function received(Request $request)
{
    $validated = $request->validate([
        'order_number' => ['required'],
    ]);

    Order::where('order_number',
$validated['order_number']->update([
        'delivery_status' => 2,
    ]);

    return response()->json([
        'success' => true,
        'message' => 'Order received successfully',
    ]);
}

public static function cancelExpiredOrder(): void
{
    $currentTime = Carbon::now();

    $expiredOrders = Order::where('expire_time', '<=',
$currentTime)
        ->where('payment_status', '=', 0)
        ->get();

    if ($expiredOrders->count() == 0) {
        return;
    }

    foreach ($expiredOrders as $expiredOrder) {
        self::cancelOrderById($expiredOrder-
>order_number);
    }
}

public static function cancelOrderById($orderNumber):
void
{
    if (!auth()->check()) {
        return;
    }

    $user = User::find(auth()->user()->id);

```

```

        DB::beginTransaction();
        try {
            $order = $user->orders()->where('order_number',
$orderNumber)->get()->firstOrFail();
            foreach ($order->orderDetails as $orderDetail)
            {
                $product = $orderDetail->product;
                $product->update([
                    'stock' => $product->stock +
$orderDetail->qty,
                ]);
            }

            $order->update([
                'payment_status' => -1,
                'delivery_status' => -1,
            ]);

            $user->notifications()->create([
                'notif_type' => '6',
                'order_number' => $order->order_number,
                'user_id' => $user->id
            ]);

            DB::commit();
        } catch (\Exception $e) {
            DB::rollBack();
        }
    }
}

```

resources/views/admin/order.blade.php

```

@extends('admin.components.sidebar', ['title' => 'Pesanan &
Transaksi', 'user' => Auth::user()])
<link rel="stylesheet" href="{{
asset('Admin/css/order.css') }}">
<link rel="stylesheet"
href="https://cdn.datatables.net/1.13.4/css/jquery.dataTabl
es.css" />

@section('content')
<div class="content-header">
    <div class="container-fluid">
        <div class="col-sm-6">
            <div class="h4">Pesanan & Transaksi</div>
        </div>
    </div>

```

```

        </div>
</div>
<section class="content">
    <div class="container-fluid">
        <div class="row mr-3 ml-3">
            <div class="category-filter mb-3">
                <select id="TableOrdersFilter" class="form-
control special">
                    <option value="">Status (Show
All)</option>
                    <option value="Dalam Pengiriman"> Dalam
Pengiriman</option>
                    <option value="Menunggu Pengiriman">
Menunggu Pengiriman</option>
                    <option value="Selesai">
Selesai</option>
                </select>
            </div>
            <div class="table-responsive">
                <table class="table display nowrap text-
center" style="width:100%" id="TableOrders">
                    <thead>
                        <tr>
                            <th scope="col">No.
Pesanan</th>
                            <th scope="col">Nama</th>
                            <th scope="col">No.
Rekening</th>
                            <th scope="col">Status
Pembayaran</th>
                            <th scope="col">Status
Pengiriman</th>
                            <th scope="col">Info</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach ($orders as $order )
                            <tr>
                                <td>{{ $order->order_number
}}</td>
                                <td>{{ $order->recipient_name
}}</td>
                                <td>{{ $order->bank-
>account_number }}</td>
                                <td class=@if ($order-
>payment_status == 1)
                                    "text-warning"

```

```

>payment_status == 2)                @elseif ($order-
                                     "text-success"
                                     @else
                                     "text-danger"
                                     @endif
                                     >{{ $order-
>payment_status_string }}</td>
                                     <td class=@if ($order-
>delivery_status == 2)
                                     "text-success"
                                     ><i class="fa fa-check-
circle-o"></i>
                                     @elseif ($order-
>delivery_status == 1)
                                     "text-warning"
                                     ><i class="fa fa-shopping-
cart"></i>
                                     @elseif ($order-
>delivery_status == 0 || ($order->receipt_number == null &&
$order->delivery_status != -1))
                                     "text-dark"
                                     ><i class="fa-regular fa-
clock"></i>
                                     @else
                                     "text-danger">
                                     @endif
                                     {{ $order-
>delivery_status_string}} </td>
                                     <td><a class="text-warning"
                                     href="{{
route('admin.order-detail', ['ordernumber'=>$order-
>order_number])}}"><i
                                     class="fa-solid fa-
circle-info"></i></a></td>
                                     </tr>
                                     @endforeach
                                     </tbody>
                                     </table>
                                     </div>
                                     </div>
                                     </div>
                                     <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jq
uery.min.js"></script>
                                     <script src="https://cdn.jsdelivr.net/npm/chartjs-
plugin-datalabels@0.7.0"></script>

```

```
<script
src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables
.js"></script>
<script>
    $("#TableOrders").dataTable({
        searching: true,
        infor: false,
        aaSorting: [],

    });

    var table = $("#TableOrders").DataTable();

    $("#TableOrdersFilter").on("change",
function(e) {
        table.column(4).search(this.value).draw();
    });
</script>
</section>
@endsection
```