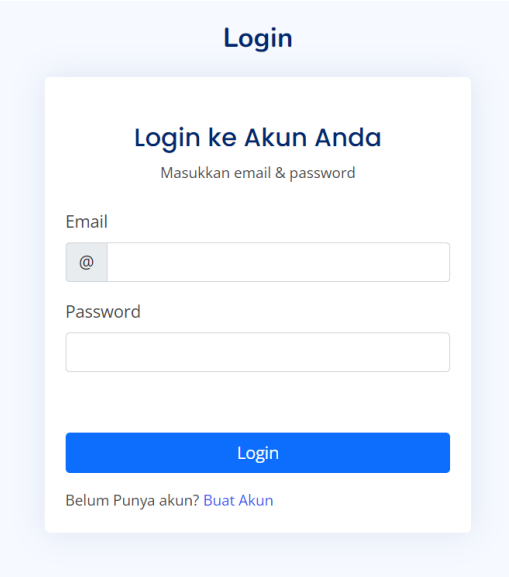


LAMPIRAN

Implementasi Proses Aplikasi

a. Implementasi Proses Login



The screenshot shows a login interface with the following elements:

- Title:** Login
- Section Header:** Login ke Akun Anda
- Instruction:** Masukkan email & password
- Form Fields:**
 - Email:** A text input field with an '@' icon on the left.
 - Password:** A text input field.
- Action:** A blue button labeled 'Login'.
- Footer:** A link that says 'Belum Punya akun? Buat Akun'.

```
class LoginController extends Controller
{
    public function login (Request $request){
        if(Auth::guard('user')->attempt(['email' => $request
            ->email, 'password' => $request->password])) ){
            return redirect('/');
        }
        elseif(Auth::guard('admin')->attempt(['email' =>
            $request->email, 'password' => $request
            ->password])) ){
            return redirect('admin/admin');
        }
        return redirect('login');
    }
    public function logout (Request $request){
        Auth::logout();
        return redirect('login');
    }
}
```

b. Implementasi Proses Registrasi User

Daftar User

Nama Anda

No. Telp Anda

Alamat Anda

Email Anda

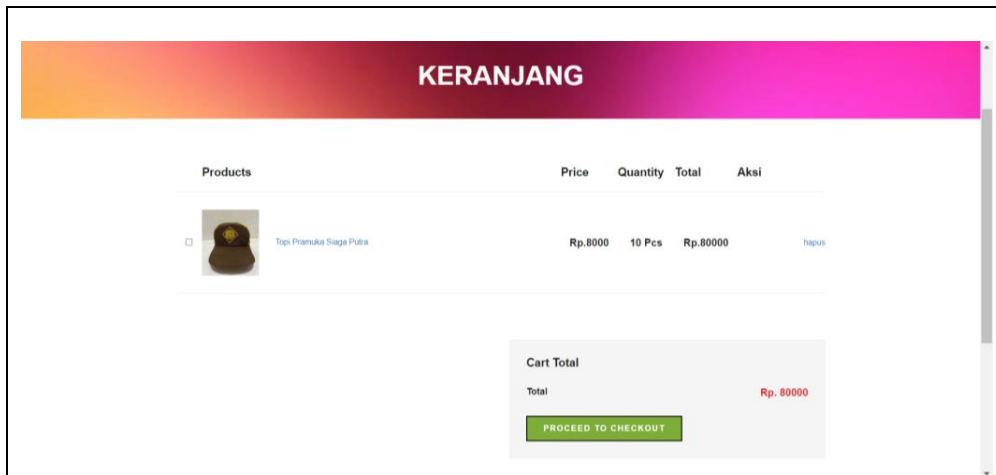
Password

[Create Account](#)

Sudah punya akun [Log in](#)

```
public function daftar(Request $request){
    DB::table('users')->insert([
        'name'=>$request->name,
        'telp'=>$request->telp,
        'alamat'=>$request->alamat,
        'email'=>$request->email,
        'password'=>bcrypt($request->pass)
    ]);
    return redirect('login');
}
```

c. Implementasi Proses Memasukkan Produk Topi Ke Keranjang



```
public function addCart(Request $request)
{
    if ($data_keranjang=DB::table('keranjang')
        ->where('id_barang',$request->id)
        ->where('id_user',auth()->user()->id)->first()){
        $data=DB::table('keranjang')
        ->where('id_barang',$request->id)->first();
        DB::table('keranjang')
        ->where('id_keranjang',$data->id_keranjang)
        ->update(['jumlah'=> $request->jmlh
        ]);
        return redirect('/keranjang');
    } else {
        DB::table('keranjang')->insert([
            'id_user'=>auth()->user()->id,
            'id_barang'=>$request->id,
            'jumlah'=>$request->jmlh
        ]);
        return redirect('/keranjang');
    }
}
```

d. Implementasi Proses Checkout

The screenshot displays a checkout interface. On the left, under 'Billing Details', there are input fields for 'Nama Penerima*' (popi), 'Telp*' (087231567088), 'Alamat Penerima*' (Jl. Melati No.07 Surabaya), 'Catatan*' (SMA Negeri Surabaya), 'Kota*' (Surabaya), and 'Kurir*' (JNE). Below these are dropdowns for 'Pilih Paket Ongkir*' and 'Layanan Reguler, Rp. 14000, 2-3 hari'. On the right, 'Your Order' section shows a table with 'Products' and 'Total' columns. The table lists 'Topi SMA Putih Abu' for Rp. 80000, a 'Subtotal' of Rp. 80000, 'Ongkir' of Rp. 14000, and a final 'Total' of Rp. 94000. Below the table is a green 'BAYAR' button and a 'Permintaan Kirim Sampel?' section with explanatory text.

```
public function payment_post(Request $request){
    $krm="";
    if(!isset($request->sampel)) {
        // echo "Not selected!";
        $krm="1";
    }else{
        //echo "Selected!";
        $krm="2";
    }
    $json = json_decode($request->get('json'));

    DB::table('transaksi')->insert([
        'id_user'=>$request->idu,
        'nama_penerima'=>$request->nama,
        'alamat_penerima'=>$request->alamat,
        'telp_penerima'=>$request->telp,
        'kurir'=>$request->kurir,
        'ongkir'=>$request->ongkir,
        'noresi'=>' - ',
        'total'=>$request->total,
        's_transaksi'=>$krm,
        'order_id'=>$json->order_id,
        's_pembayaran'=>$json->transaction_status,
        'catatan'=>$request->catatan
    ]);
    $orderID=DB::table('transaksi')
    ->where('id_user', '=', auth()->user()->id)
```

```

->orderBy('id_transaksi', 'desc')->first();
foreach(session('cart') as $id => $details){
    DB::table('detail_transaksi')->insert([
        'id_transaksi'=>$orderID->id_transaksi,
        'id_barang'=>$details['idb'],
        'jumlah_barang'=>$details['banyak']
    ]);
}
foreach(session('cart') as $id => $details){
    DB::table('keranjang')
->where('id_barang',$details['idb'])
->where('id_user',auth()->user()->id)
->delete();
}
session()->forget('cart');
return redirect('/');
}

```

e. Implementasi Proses Menambahkan Produk

ID	Nama Produk	Gambar	Jumlah	Harga	Aksi
17	Topi SMP Biru		100	Rp. 8000	Edit - Hapus
18	Topi SMP Biru Putih		100	Rp. 8000	Edit - Hapus
19	Topi SMA Full Abu		100	Rp. 8000	Edit - Hapus
20	Topi SMA Putih Abu		100	Rp. 8000	Edit - Hapus
25	Topi Sekolah Dasar uji coba		100	Rp. 8000	Edit - Hapus

```

public function tambah(){
    $data_jenis = DB::table('jenis_barang')->get();
    return
view('admin.barang.barang_tambah',['data_jenis'=>$data_jenis
]);
}
public function simpan(Request $request){
    $this->validate($request,[

```

```

        'gambar' =>
        'required|mimes:png,jpg,jpeg|max:2048',
    ]);
    $file=$request->file('gambar');
    $namabaruu = str_replace(' ', '', $request->nama);
    $nama_file=$namabaruu.'.'.$file
    ->getClientOriginalExtension();

    $tujuan='image/';
    $file->move($tujuan,$nama_file);
    DB::table('barang')->insert([
        'nama_barang'=>$request->nama,
        'gambar_barang'=>$nama_file,
        'jumlah_barang'=>$request->jumlah,
        'harga_barang'=>$request->harga,
        'ket_barang'=>$request->ket,
        'id_jenis'=>$request->jenis,
        's_barang'=>'1'
    ]);
    return redirect('admin/barang');
}

```

f. Implementasi Proses Menambahkan Kategori Produk Topi

```

public function tambah(){
    return view('admin.jenis.jenis_tambah');
}
public function simpan(Request $request){
    DB::table('jenis_barang')->insert([

```

```

        'nama_jenis'=>$request->nama,
        's_jenis'=>'1'
    ]);
    return redirect('admin/jenis');
}

```

g. Implementasi Proses Mengupdate Nomor Resi

```

public function updateresi(Request $request){
    DB::table('transaksi')
    ->where('id_transaksi',$request->id)->update([
        'noresi'=>$request->resi
    ]);
    return redirect('transaksi/transaksi');
}

```

h. Implementasi Proses Menambahkan Admin

```

public function tambah(){
    return view('admin.admin.admin_tambah');
}
public function simpan(Request $request){

```

```

DB::table('admins')->insert([
    'name'=>$request->nama,
    'email'=>$request->email,
    'password'=>bcrypt($request->pass)
]);
return redirect('admin/admin');
}

```

i. Implementasi Proses Admin Mengupdate Status

The screenshot shows an Admin dashboard with a sidebar menu containing 'Dashboard', 'Transaksi', 'Data Master', and 'Logout'. The main content area is titled 'Data Transaksi' and displays a table of transactions. The table has columns for '#', 'Order Id', 'Nama Pemesan', 'Alamat Pemesan', 'Pengiriman', 'Total', 'Status', and 'Aksi'. There are five rows of data, each with a corresponding 'Update Resi' and 'Sampel Dikirim' button in the 'Aksi' column.

#	Order Id	Nama Pemesan	Alamat Pemesan	Pengiriman	Total	Status	Aksi
53	666289845	aida	Jl. soehat	jne	174000		
52	242740495			jne	188000	Dana Terkirim	
51	475219442	joji	Jl. Kapuas, Aceh	jne	172000	Kirim Sampel	Sampel Dikirim
50	475219442	joji	Jl. Kapuas, Aceh	jne	172000	Kirim Sampel	Update Resi - Sampel Dikirim
49	977546955	joji	Jl. Kapuas, Aceh	jne	246000	Kirim Sampel	Update Resi - Sampel Dikirim

```

public function updatestatus(Request $request){
    DB::table('transaksi')-
    >where('id_transaksi',$request->id)->update([
        's_transaksi'=>$request->status
    ]);
    return redirect('transaksi/transaksi');
}

```