

LAPORAN TUGAS AKHIR
RANCANG BANGUN SISTEM INFORMASI
PEMBELIAN DAN PENJUALAN BERBASIS WEBSITE
PADA ADDICTIVE CLOTHING STORE



Dibuat untuk Memenuhi Syarat Menyelesaikan Tugas Akhir
Semester IV di Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Jember

Disusun Oleh:

Arfan Astaraja	E41212007
Dinda Amalia Julyandri	E41212015
Bilal Shandyarta Syamsudin	E41212037
Lya Nurul Ula	E41211962

PROGRAM STUDI TEKNIK INFORMATIKA PSDKU SIDOARJO

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2023

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Puji syukur kita panjatkan kehadiran Allah Swt. yang telah memberikan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan tugas akhir yang berjudul "Rancang Bangun Sistem Informasi Pembelian dan Penjualan Berbasis Website pada Addictive Clothing Store" ini tepat pada waktunya. Adapun tujuan dari penulisan laporan ini adalah untuk memenuhi tugas syarat menyelesaikan tugas akhir semester IV di Program Studi D-IV Teknik Informatika

Terlebih dahulu, saya mengucapkan terima kasih kepada bapak atau ibu dosen pengampu mata kuliah Workshop Sistem Informasi Berbasis Web, dan Workshop Sistem Informasi Berbasis Mobile. yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan bidang studi yang saya tekuni ini. kami juga mengucapkan terima kasih kepada semua pihak yang tidak dapat saya sebutkan semua, terima kasih atas bantuannya sehingga saya dapat menyelesaikan tugas ini.

Kemudian, kami menyadari bahwa tugas yang kami tulis ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun kami butuhkan demi kesempurnaan laporan ini.

Wassalamu álaikum warahmatullahi wabarakatuh

Sidoarjo, 12 Juni 2023

Penyusun

DAFTAR PUSATAKA

SAMPUL	1
KATA PENGANTAR	2
DAFTAR PUSATAKA	3
BAB 1 PENDAHULUAN	5
1.1 Latar Belakang	5
1.2 Rumusan Masalah	5
1.3 Tujuan	5
1.4 Batasan Masalah	5
BAB 2 TINJAUAN PUSTAKA	6
2.1 <i>Website</i>	6
2.2 <i>Database</i>	6
2.3 <i>E-Commerce</i>	6
BAB III	7
3.1 <i>Design System</i>	7
3.2 <i>Flowchart</i>	8
3.3 ERD	9
BAB 4 HASIL DAN PEMBAHASAN	11
4.1 <i>Source Code</i>	11
4.1.1 <i>Login</i>	11
4.1.2 <i>Register</i>	13
4.1.3 <i>Home</i>	15
4.1.4 <i>Detail</i>	18
4.1.5 <i>Favorite</i>	21
4.1.6 <i>Checkout</i>	24
4.1.7 <i>Pesanan Anda</i>	28
4.1.8 <i>Profile</i>	29
4.1.9 <i>Ubah Profile</i>	32
4.1.10 <i>Admin Input Barang</i>	35
4.1.11 <i>Admin Barang</i>	37
4.1.12 <i>Edit Barang</i>	40
4.1.13 <i>Admin User</i>	42

4.1.14 Admin Pesanan.....	44
4.1.15 Konfirmasi Pesanan	45
BAB 5 KESIMPULAN DAN SARAN	48
5.1 Kesimpulan	48
5.2 Saran	48
DAFTAR PUSTAKA.....	49

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada saat ini penjualan berbasis *e-commerce* saat udah tidak asing lagi di kalangan masyarakat hingga perusahaan perdagangan. Hal ini disebabkan oleh banyaknya pedagang maupun perusahaan yang menganggap dengan adanya fitur *e-commerce* dapat memberikan keuntungan secara finansial maupun non finansial (Susandi & Sukisno, 2017). Kegiatan jual beli yang dilakukan menggunakan teknologi yang maju kini menjadi senjata untuk tetap bertahan dan memenangkan persaingan yang semakin hari semakin ketat. Dengan adanya *website* toko *online* yang dapat memudahkan pelaku usaha atau toko untuk mempromosikan produknya dan mempermudah konsumen untuk mendapatkan informasi tentang produk-produk yang dimiliki penjual atau toko (Susanti & Nurdiana, 2018).

Dalam kasus ini Addictive Clothing merupakan UMKM yang bergerak di bidang penjualan produk kaos distro dengan desain yang unik dan menarik. Addictive Clothing Store didirikan sejak tahun 2021 oleh beberapa remaja yang berasal dari Kediri. Selain menjual pakaian distro, Addictive Clothing Store juga siap melayani pemesanan seperti kaos lusinan, kaos jersey, sablon DTF, sablon DTG, dan juga sablon plastisol. Sistem penjualan, pemasaran dan promosi yang dipakai saat ini masih menggunakan sistem secara manual. Maka dari itu untuk membantu dan mempermudah transaksi jual beli pada Addictive Clothing diperlukan sebuah website yang menyediakan katalog produk yang berisi gambar dan informasi produk yang dapat menjadi sarana bagi konsumen untuk mengirimkan desain pakain yang ingin mereka pesan (Hastanti & Purnama, 2015).

1.2 Rumusan Masalah

Dari latar belakang diatas permasalahan yang menjadi dasar pembuatan Tugas Akhir ini adalah “Bagaimana merancang sebuah sistem penjualan kaos berbasis web pada Addictive Clothing Store agar dapat membantu dan mempermudah transaksi jual beli dan lebih dikenal oleh masyarakat luas”.

1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk “Merancang sebuah sistem informasi penjualan komputer secara online dengan berbasis web agar dapat membantu dan mempermudah transaksi jual beli dan lebih dikenal oleh masyarakat luas.

1.4 Batasan Masalah

Untuk menghindari pembuatan laporan yang tidak terarah maka penyusun membuat batasan masalah. Adapun masalah yang dibahas dalam laporan ini hanya membahas tentang pemesanan produk pada Addictive clothing Store menggunakan website.

BAB 2 TINJAUAN PUSTAKA

2.1 Website

Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet (Abdulloh, 2016). *Website* juga dapat diartikan sebagai sebuah halaman yang berisi data, baik data text, gambar, suara dan lainnya yang dapat diakses secara online. ada banyak model pengembangan sistem yang bisa dimanfaatkan untuk membangun website salah satunya model Prototyping. Prototyping merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dikehendakinya.

2.2 Database

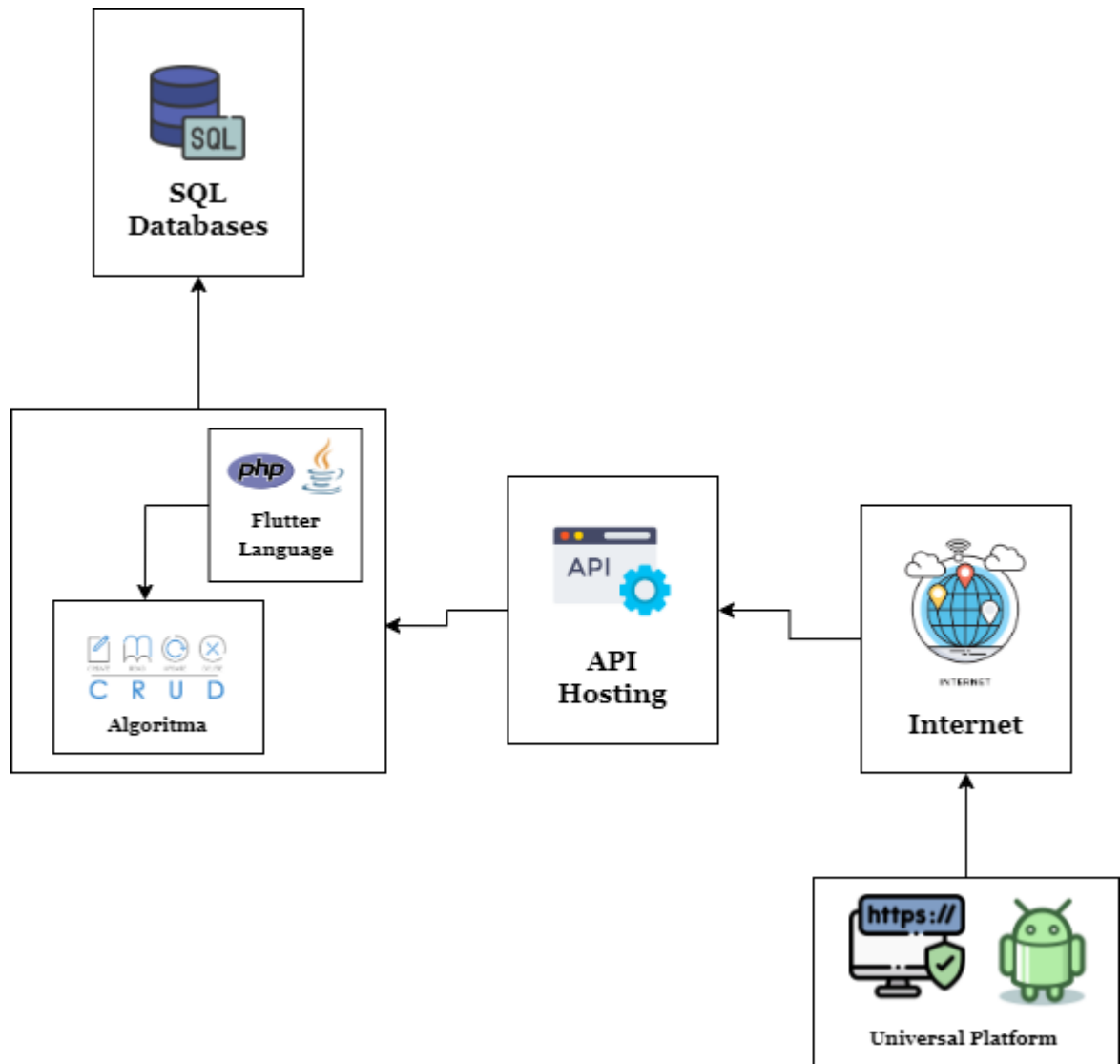
Database atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Kegunaan utama sistem basis data adalah agar pemakai mampu menyusun suatu pandangan (*view*) abstraksi data. Hal ini bertujuan untuk menyederhanakan interaksi antara pengguna dengan sistemnya dan basis data dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer, dan administratornya (Andaru, 2018).

2.3 E-Commerce

E-Commerce merupakan suatu sistem atau paradigma baru dalam dunia bisnis, yang menggeser paradigma perdagangan tradisional menjadi *electronic commerce* yaitu dengan memanfaatkan teknologi ICT (*Information and Communication Technology*), atau dengan kata lain teknologi internet. Definisi *e-commerce* secara umum : “ Proses membeli, menjual, baik dalam bentuk barang, jasa ataupun informasi, yang dilakukan melalui media internet”. Menurut Stefan Probst (Opticom), definisi *e-commerce* adalah “Bisnis yang dilakukan secara *electronic* yang melibatkan aktivitas-aktivitas bisnis berupa *business to business* ataupun *business to konsumen* melalui teknologi internet.” *E-business* adalah transaksi yang menggunakan media elektronik yang dipergunakan untuk berjualan atau proses pembelian atau proses pembelian suatu atau beberapa produk menggunakan teknologi ICT. Secara umum, interaksi dan transaksi antara pelaku bisnis yang akan menggunakan teknologi *e-commerce* dapat dikategorikan dalam jenis B2B (*business to business*), B2C (*business to konsumen*), C2B (*konsumen to business*), dan C2C (*konsumen to konsumen*) (Yawedani & Wijaya, 2017).

BAB III

3.1 Design System



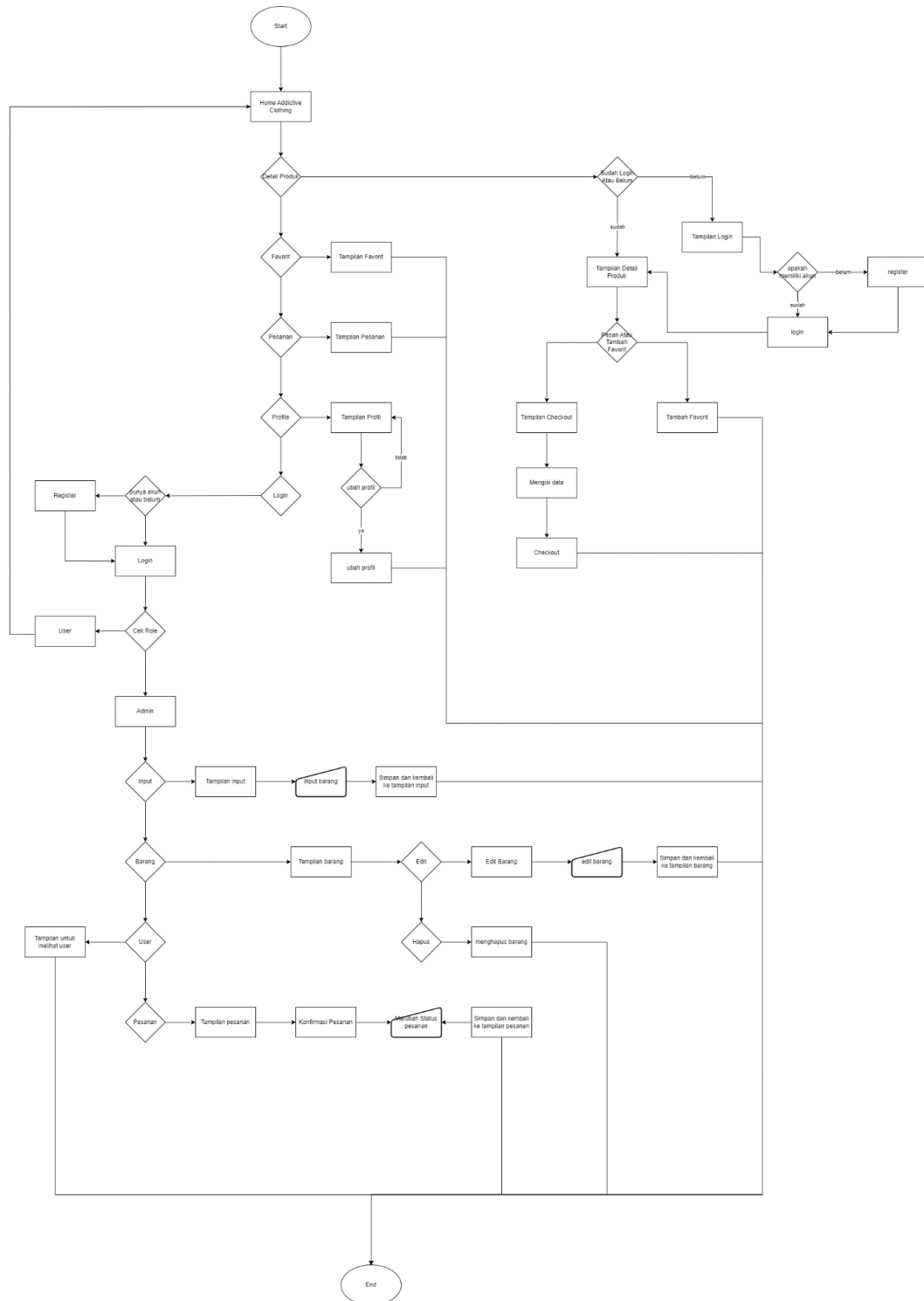
Sistem desain untuk Addictive Clothing memiliki alur sebagai berikut:

Website dan Aplikasi Mobile terhubung ke internet dan mengambil API untuk mencapai integrasi antara keduanya. Hal ini memungkinkan pengguna untuk mengakses fitur dan data yang sama melalui platform yang berbeda. API Hosting berperan sebagai penghubung antara Website dan Aplikasi Mobile dengan menghasilkan algoritma CRUD (Create, Read, Update, Delete) yang terintegrasi dalam Aplikasi Mobile yang dikembangkan menggunakan Flutter. Algoritma CRUD ini memungkinkan pengguna untuk melakukan operasi dasar seperti membuat, membaca, memperbarui, dan menghapus data melalui aplikasi mobile. Namun untuk saat ini fitur memperbarui dan mengupdate data hanya dapat dilakukan pada admin website.

Hasil dari operasi CRUD yang dilakukan melalui admin pada Website akan mengubah data yang ada dalam SQL database. Ini berarti bahwa ketika admin melakukan perubahan data melalui antarmuka admin di Website, data yang terkait dalam database SQL akan diperbarui sesuai dengan perubahan yang dilakukan.

Dengan alur ini, sistem Addictive Clothing menciptakan keterhubungan antara *Website*, Aplikasi Mobile, API Hosting, dan SQL *database*, memastikan integritas data dan pengalaman pengguna yang konsisten di berbagai platform.

3.2 Flowchart



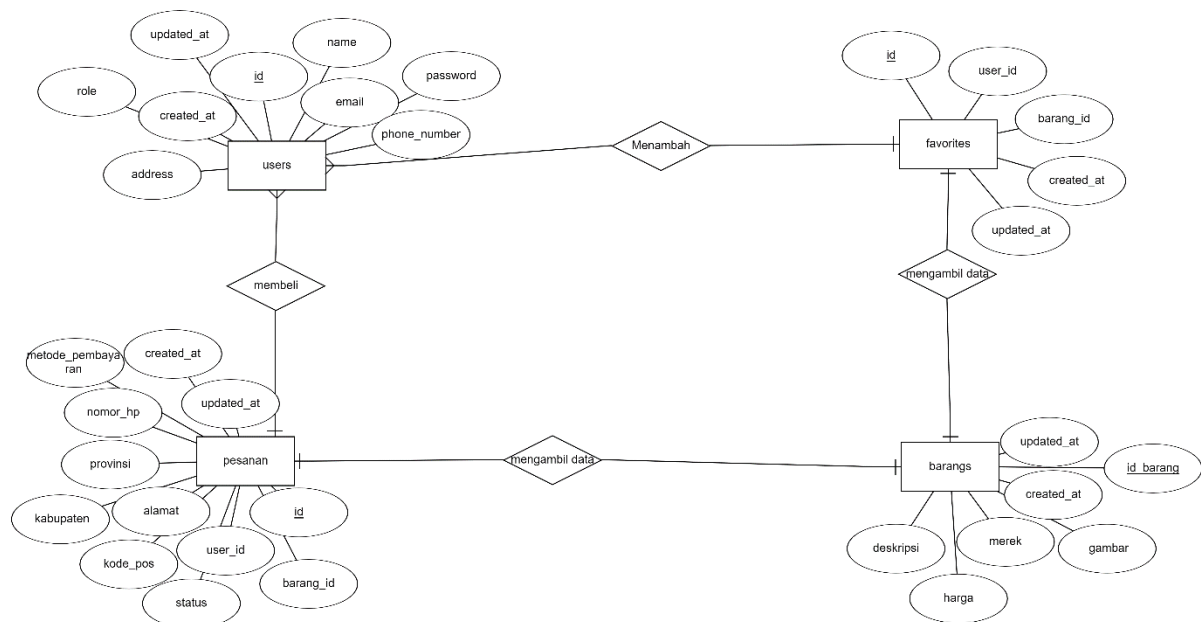
Dalam website addictive clothing, terdapat dua peran pengguna yaitu admin dan pengguna biasa. Setelah melakukan login, pengguna akan diarahkan ke halaman beranda jika memiliki peran "user", sedangkan jika memiliki peran "admin", pengguna akan

diarahkan ke halaman admin. Untuk melihat detail produk, pengguna harus login terlebih dahulu. Setelah masuk ke halaman detail produk, pengguna dapat melakukan pemesanan dengan memilih tombol "pesan" dan akan diarahkan ke halaman checkout. Setelah proses checkout selesai, pengguna akan diarahkan ke halaman pesanan yang berisi daftar semua pesanan yang telah mereka buat.

Pada peran admin, admin memiliki beberapa fitur yang dapat dilakukan, antara lain:

- Memasukkan barang : Admin dapat menambahkan produk baru ke dalam sistem dengan mengisi informasi yang diperlukan, seperti merek, deskripsi, harga, dan gambar.
- Mengedit barang : Admin dapat mengubah informasi barang yang sudah ada, seperti mengganti gambar, mengedit deskripsi, atau mengubah harga.
- Menghapus barang : Admin dapat menghapus barang yang tidak lagi tersedia atau tidak relevan.
- Melihat pengguna : Admin dapat melihat daftar pengguna yang terdaftar dalam sistem, termasuk informasi seperti nama, alamat email, dan detail kontak lainnya.
- Melihat pesanan : Admin dapat melihat daftar pesanan yang telah dibuat oleh pengguna. Informasi yang dapat dilihat meliputi detail pesanan, informasi pengiriman, dan status pembayaran.
- Mengonfirmasi pesanan : Admin dapat mengonfirmasi pesanan yang telah dibuat oleh pengguna, mengubah status pesanan, dan memberikan pembaruan kepada pengguna mengenai status pesanan mereka.

3.3 ERD



Dalam ERD yang dibuat untuk sistem Addictive Clothing, terdapat empat entitas utama: *users*, *barang*, *favorites*, dan *pesanans*. Setiap entitas memiliki atribut-atribut yang relevan dengan fungsinya.

Entitas *users* menyimpan informasi penting tentang pengguna, seperti ID, nama, alamat email, nomor telepon, dan alamat. Atribut *role* menunjukkan peran pengguna dalam sistem, sedangkan atribut *created_at* dan *updated_at* menyimpan informasi tentang tanggal dan waktu pembuatan serta pembaruan data pengguna.

Entitas *barang* menyimpan informasi tentang barang-barang yang tersedia di dalam sistem. Atribut *id_barang* digunakan sebagai identifier unik untuk setiap barang, sementara atribut *gambar*, *merek*, *harga*, dan *deskripsi* memberikan detail tentang barang tersebut. Atribut *created_at* dan *updated_at* mencatat tanggal dan waktu pembuatan serta pembaruan data barang.

Entitas *favorites* menghubungkan pengguna dengan barang-barang yang disukai. Atribut *id*, *user_id*, dan *barang_id* digunakan untuk mengaitkan pengguna dengan barang yang menjadi favorit mereka. Atribut *created_at* dan *updated_at* mencatat informasi tentang tanggal dan waktu penambahan barang ke dalam daftar favorit.

Entitas *pesanan* menyimpan informasi tentang pesanan yang dibuat oleh pengguna. Atribut *id*, *user_id*, dan *barang_id* menghubungkan pesanan dengan pengguna dan barang yang dipesan. Atribut *status* menunjukkan status pesanan, sedangkan atribut lainnya, seperti *alamat*, *provinsi*, *kabupaten*, *kode_pos*, *metode_pembayaran*, *nomor_hp*, *created_at*, dan *updated_at*, berisi detail terkait pengiriman dan informasi pembayaran.

ERD ini memvisualisasikan hubungan antara entitas-entitas tersebut dan atribut-atribut yang menyertainya, membantu memahami struktur dan fungsionalitas sistem *addictive clothing*.

BAB 4 HASIL DAN PEMBAHASAN

4.1 Source Code

4.1.1 Login

- Fungsi untuk menampilkan View login

```
public function showLoginForm()
{
    return view('akun.login');
}
```

Baris ini untuk menampilkan file yang berada di resources/views/akun/login

```
return view('akun.login');
```

- Fungsi Login

```
public function login(Request $request)
{
    $credentials = $request->validate([
        'email' => 'required|email',
        'password' => 'required',
    ], [
        'email.required' => 'Email harus diisi.',
    ]);

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

        if (Auth::user()->role === 'admin') {
            return redirect()->intended('/input');
        } else {
            return redirect()->intended('/');
        }
    }

    return back()->withInput()->withErrors([
        'email' => 'Email atau password salah.',
    ]);
}
```

Baris ini memvalidasi data yang diterima dari permintaan (\$request). Validasi dilakukan menggunakan aturan berikut: kolom 'email' harus diisi dan harus berupa alamat email yang valid, serta kolom 'password' harus diisi. Jika validasi gagal, pesan kesalahan akan ditampilkan dengan menggunakan pesan khusus 'Email harus diisi.'

```
$credentials = $request->validate([
    'email' => 'required|email',
    'password' => 'required',
], [
    'email.required' => 'Email harus diisi.',
]);
```

Baris ini melakukan percobaan otentikasi dengan menggunakan kredensial yang validasi sebelumnya. Jika otentikasi berhasil, sesi akan diregenerasi dan dilakukan pengecekan peran pengguna yang berhasil masuk. Jika peran pengguna adalah 'admin', pengguna akan diarahkan ke '/input', jika bukan, pengguna akan diarahkan ke halaman utama ('/').

```

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

    if (Auth::user()->role === 'admin') {
        return redirect()->intended('/input');
    } else {
        return redirect()->intended('/');
    }
}

```

Baris ini akan dijalankan jika otentikasi gagal. Fungsi back() akan mengarahkan pengguna kembali ke halaman sebelumnya. Metode withInput() akan mengisi kembali input pengguna sebelumnya agar tidak hilang. Metode withErrors() akan menambahkan pesan kesalahan 'Email atau password salah.' ke dalam sesi flash untuk ditampilkan kepada pengguna.

```

return back()->withInput()->withErrors([
    'email' => 'Email atau password salah.',
]);

```

- **Fungsi Logout**

```

public function logout(Request $request)
{
    Auth::logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/');
}

```

Baris ini memanggil fungsi logout dari fasilitas otentikasi Laravel. Fungsi ini akan menghapus sesi otentikasi pengguna saat ini.

```
Auth::logout();
```

Baris ini menghapus sesi yang terkait dengan permintaan saat ini. Hal ini berguna untuk menghancurkan sesi pengguna yang terkait dengan otentikasi.

```
$request->session()->invalidate();
```

Baris ini menghasilkan dan mengatur ulang token sesi untuk melindungi aplikasi dari serangan perusakan sesi. Dengan mengatur ulang token sesi, serangan perusakan sesi akan sulit dilakukan.

```
$request->session()->regenerateToken();
```

Pada baris ini mengarahkan pengguna ke halaman utama ('/') setelah proses logout selesai.

```
return redirect('/');
```

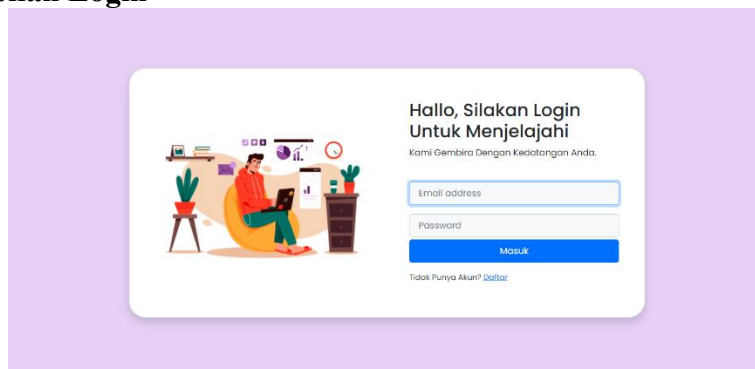
- **Source Code View Login**

```

@extends('akun.layouts.base')
@section('login')
<div class="col-md-6 right-box">
<div class="row align-items-center">
<div class="header-text mb-4">
<h2>Hallo, Silakan Login Untuk Menjelajahi</h2>
<p>Kami Gembira Dengan Kedatangan Anda.</p>
</div>
<form method="POST" action="{{ route('login') }}">
@csrf
<div class="input-group mb-3">
<input type="email" class="form-control form-control-lg bg-light fs-6" placeholder="Email address"
name="email" value="{{ old('email') }}" required autocomplete="email" autofocus>
</div>
<div class="input-group mb-1">
<input type="password" class="form-control form-control-lg bg-light fs-6" placeholder="Password"
name="password" required autocomplete="current-password">
</div>
@if ($errors->any())
<div class="alert alert-danger" role="alert">
{{ $errors->first() }}
</div>
@endif
<div class="input-group mb-3">
<button class="btn btn-lg btn-primary w-100 fs-6" type="submit">Masuk</button>
</div>
<div class="row">
<small>Tidak Punya Akun? <a href="{{ route('register.index') }}">Daftar</a></small>
</div>
</form>
</div>
</div>
@endsection

```

- **Tampilan Login**



4.1.2 Register

- **Fungsi Menampilkan Halaman Register**

```

public function index(){
    return view('akun.register');
}

```

Pada baris ini untuk menampilkan file view yang berada di direktori resources/views/akun/register

```

return view('akun.register');

```

- **Fungsi Register**

```

public function store(Request $request){
    $request->validate([
        'name' => 'required',
        'email' => [
            'required',
            'email',
            Rule::unique('users')
        ],
        'password' => 'required|min:8|confirmed',
        'password_confirmation' => 'required'
    ]);

    $user = new User;
    $user->name = $request->name;
    $user->email = $request->email;
    $user->password = Hash::make($request->password);
    $user->save();

    return redirect('/login')->with('success', 'Akun berhasil dibuat. Silakan login.');
```

Baris ini memvalidasi data yang diterima dari permintaan (\$request). Validasi dilakukan menggunakan aturan berikut: kolom 'name' harus diisi, kolom 'email' harus diisi dan harus berupa alamat email yang valid, kolom 'email' harus unik dalam tabel 'users', kolom 'password' harus diisi dan minimal terdiri dari 8 karakter, serta kolom 'password_confirmation' harus diisi. Jika validasi gagal, Laravel akan secara otomatis mengembalikan respons dengan pesan kesalahan yang sesuai.

```

$request->validate([
    'name' => 'required',
    'email' => [
        'required',
        'email',
        Rule::unique('users')
    ],
    'password' => 'required|min:8|confirmed',
    'password_confirmation' => 'required'
]);
```

Baris ini membuat objek User baru dan mengisi propertinya dengan data yang diterima dari permintaan. Password juga di-hash menggunakan fungsi Hash::make() sebelum disimpan ke database. Kemudian, objek User tersebut disimpan ke dalam database.

```

$user = new User;
$user->name = $request->name;
$user->email = $request->email;
$user->password = Hash::make($request->password);
$user->save();
```

Baris ini mengarahkan pengguna ke halaman login setelah berhasil membuat akun. Metode with() digunakan untuk menyimpan pesan sukses 'Akun berhasil dibuat. Silakan login.' ke dalam sesi flash. Pesan ini dapat ditampilkan setelah pengguna diarahkan ke halaman login.

```

return redirect('/login')->with('success', 'Akun berhasil dibuat. Silakan login.');
```

- [Source Code View Register](#)

```

@extends('akun.layouts.base')
@section('register')
    <div class="col-md-6 right-box">
        <div class="row align-items-center">
            <div class="header-text mb-4">
                <h2>Hallo, Mohon Untuk Daftar Terlebih Dahulu</h2>
                <p>Kami Sangat Menyambut Anda.</p>
            </div>
            <form method="POST" action="{{ route('register.store') }}">
                @csrf
                <div class="input-group mb-3">
                    <input type="text" name="name" class="form-control form-control-lg bg-light fs-6" placeholder="Nama" value="{{ old('name') }}" required autocomplete="name" autofocus>
                    @error('name')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
                <div class="input-group mb-3">
                    <input type="email" name="email" class="form-control form-control-lg bg-light fs-6" placeholder="Email" value="{{ old('email') }}" required autocomplete="email">
                    @error('email')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
                <div class="input-group mb-3">
                    <input type="password" name="password" class="form-control form-control-lg bg-light fs-6" placeholder="Password" required autocomplete="new-password">
                    @error('password')
                        <span class="invalid-feedback" role="alert">
                            <strong>{{ $message }}</strong>
                        </span>
                    @enderror
                </div>
                <div class="input-group mb-1">
                    <input type="password" name="password_confirmation" class="form-control form-control-lg bg-light fs-6" placeholder="Konfirmasi Password" required autocomplete="new-password">
                </div>
                <div class="boxsized"></div>
                <div class="input-group mb-3">
                    <button type="submit" class="btn btn-lg btn-primary w-100 fs-6">Daftar</button>
                </div>
                <div class="row">
                    <small>Sudah Punya Akun? <a href="{{ route('login') }}">Masuk</a></small>
                </div>
            </form>
        </div>
    </div>
@endsection

```

- **Tampilan Register**

4.1.3 Home

- **Fungsi Menampilkan Home**

```

public function index(){
    $barangs = Barang::all();
    return view('home.home', compact('barangs'));
}

```

Baris ini mengambil semua data barang dari model Barang dan menyimpannya ke dalam variabel \$barangs. Fungsi all() digunakan untuk mengambil semua data dari tabel barangs. Baris ini

```
$barangs = Barang::all();
```

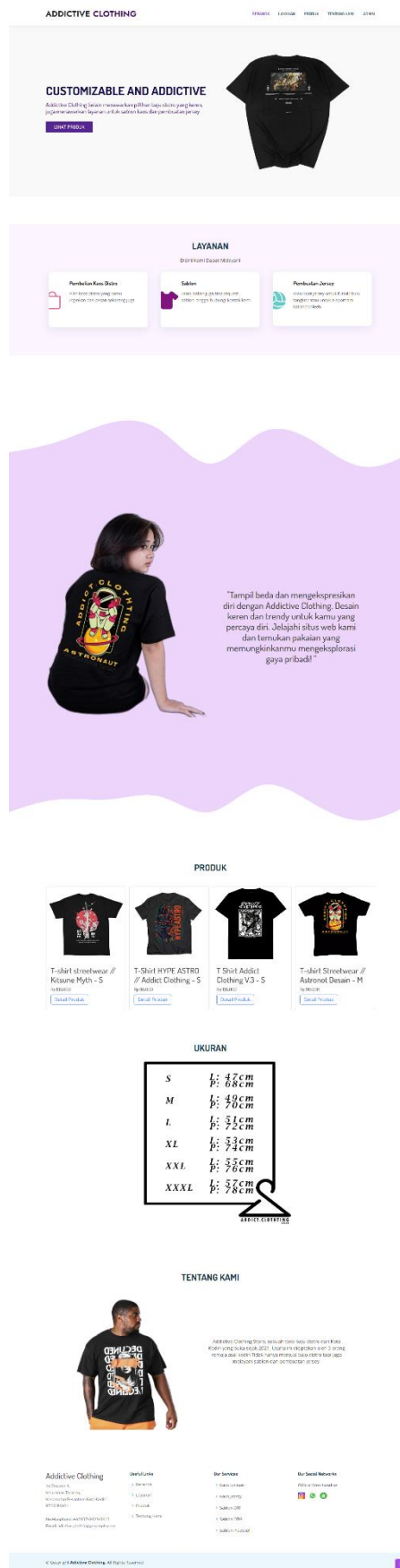
Baris ini menampilkan file view dari direktori resources/views/home/home. Tampilan ini akan diisi dengan data yang dikirimkan melalui parameter compact('barangs'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$barangs akan tersedia di tampilan dengan nama yang sama.

```
return view('home.home', compact('barangs'));
```

- **Source Code View Home**

[illegible]

- **Tampilan Home**



4.1.4 Detail

- **Fungsi Menampilkan Halaman Detail**

```
public function show($id_barang){  
    if (Auth::check()) {  
        $barang = Barang::where('id_barang', $id_barang)->first();  
        return view('details.detail', compact('barang'));  
    } else {  
        return view('akun.login');  
    }  
}
```

Baris ini memeriksa apakah pengguna sedang terotentikasi atau tidak menggunakan fungsi Auth::check(). Jika pengguna sedang terotentikasi, maka kode dalam blok if akan dieksekusi.

```
if (Auth::check()) {
```

Baris ini mengambil data barang dari tabel barang berdasarkan ID barang yang diberikan. Data barang tersebut kemudian dikirimkan ke tampilan (view) dengan nama 'details.detail' menggunakan fungsi view(). Variabel \$barang akan tersedia di tampilan dengan nama yang sama berkat penggunaan compact('barang').

```
$barang = Barang::where('id_barang', $id_barang)->first();  
return view('details.detail', compact('barang'));
```

Baris ini akan dieksekusi jika pengguna tidak terotentikasi. Jika demikian, pengguna akan diarahkan ke halaman login dengan menggunakan tampilan (view) dengan nama 'akun.login'.

```
} else {  
    return view('akun.login');  
}
```

- **Source Code View**

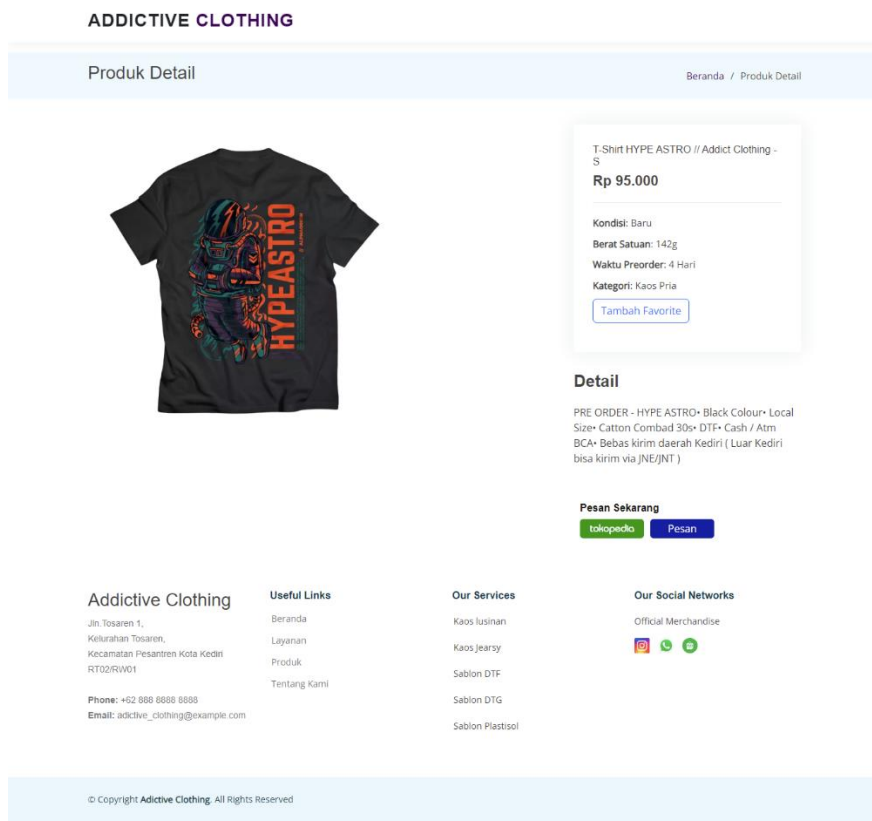
```

@extends('details.layouts.base')
@section('content')
    <main id="main">
        <section id="breadcrumbs" class="breadcrumbs">
            <div class="container">
                <div class="d-flex justify-content-between align-items-center">
                    <h2>Produk Detail</h2>
                    <ol>
                        <li><a href="{{ url('/') }}">Beranda</a></li>
                        <li>Produk Detail</li>
                    </ol>
                </div>
            </div>
        </section>
        <section id="portfolio-details" class="portfolio-details">
            <div class="container">
                <div class="row gy-4">
                    <div class="col-lg-8 id="gmbr">
                        
                    </div>
                    <div class="col-lg-4">
                        <div class="portfolio-info">
                            <h6>{{ $barang->merek }}</h6>
                            <h3>Rp {{ $barang->harga }}</h3>
                            <ul>
                                <li><strong>Kondisi</strong>: Baru</li>
                                <li><strong>Berat Satuan</strong>: 142g</li>
                                <li><strong>Waktu Preorder</strong>: 4 Hari</li>
                                <li><strong>Kategori</strong>: Kaos Pria</li>
                                <li><strong>
                                    <!-- Tombol favorit hanya ditampilkan saat pengguna sudah Login -->
                                    @auth
                                        <a href="{{ route('add.favorite', $barang->id_barang) }}"
                                            class="btn btn-outline-primary">Tambah Favorite</a>
                                    @endauth

                                    <!-- Jika pengguna belum Login, tombol favorit akan mengarahkan ke halaman Login -->
                                    @guest
                                        <a href="{{ route('login') }}" class="btn btn-outline-primary">
                                            Login untuk
                                            Menambahkan ke Favorit</a>
                                    @endguest
                                </strong></li>
                            </ul>
                        </div>
                        <div class="portfolio-description">
                            <h2>Detail</h2>
                            <p>{!! nl2br(e($barang->deskripsi)) !!}</p>
                        </div>
                    </div>
                </div>
                <div class="pesan">
                    <h6 id="pesan">Pesan Sekarang</h6>
                    <div class="tombol">
                        <div class="logo-tokped">
                            <a href=
                                "https://www.tokopedia.com/addictclothing?utm_source=sellerchannel&utm_campaign=Shop-0-14067566-0&utm_medium=share&fbclid=PAAab0N5qm5ms27FD8xXXWeHo1C169n8onv5jJeOMYMHZGhwTRGbrw13KtaXk&branch_match_id=1173616213403730326&branch_referrer=H4sIAAAAAAAsokSkottXL8nPzi9ITc1M1MvJzMvWt0xJyUwuSc7JL8koycXlt09LSs7JTLEncHRMTDLwMy3MwC0tNjJ3c7GoiiGIT/XIz3HOMbPMs8jPKzPN8kr194309VhyzygPCXJPCio3NPYOSYzIBgB7mMr2awAAAA="
                                target="blank"><span class="btn-logo">tokopedia</span></a>
                        </div>
                        <div class="btn-psn">
                            <a href="{{ route('checkout', $barang->id_barang) }}">
                                <span class="btn-pesan">Pesan</span>
                            </a>
                        </div>
                    </div>
                </div>
            </div>
        </section>
    </main>
@endsection

```

- **Tampilan Detail**



4.1.5 Favorite

- **Fungsi Menampilkan Halaman Favorite**

```
public function index()
{
    $userId = Auth::id();
    $favorites = Favorite::with('barang')
        ->where('user_id', $userId)
        ->get();
    return view('favorite.favorite', compact('favorites'));
}
```

Baris ini mendapatkan ID pengguna yang sedang terotentikasi saat ini dengan menggunakan fungsi `Auth::id()`. ID pengguna ini akan digunakan dalam pengambilan data dari tabel favorit.

```
$userId = Auth::id();
```

Baris ini mengambil data favorit pengguna yang terkait dengan ID pengguna yang sedang terotentikasi. Fungsi `with('barang')` digunakan untuk memuat relasi 'barang' yang terhubung dengan model Favorite. Metode `where('user_id', $userId)` digunakan untuk memfilter data favorit berdasarkan ID pengguna. Metode `get()` digunakan untuk mendapatkan kumpulan data favorit yang memenuhi kondisi tersebut.

```
$favorites = Favorite::with('barang')
    ->where('user_id', $userId)
    ->get();
return view('favorite.favorite', compact('favorites'));
```

Baris ini menampilkan file view dari direktori resources/views/favorite/favorite. Tampilan ini akan diisi dengan data yang dikirimkan melalui parameter compact('favorites'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$favorites akan tersedia di tampilan dengan nama yang sama.

```
return view('favorite.favorite', compact('favorites'));
```

- **Fungsi Menambahkan Favorite**

```
public function addtoFavorite(Request $request){
    $barangId = $request->route('id');

    $existingFavorite = Favorite::where('user_id', auth()->id())
        ->where('barang_id', $barangId)
        ->first();

    if (!$existingFavorite) {
        $favorite = new Favorite();
        $favorite->user_id = auth()->id();
        $favorite->barang_id = $barangId;
        $favorite->save();

        return redirect()->back()->with('success', 'Barang berhasil ditambahkan ke favorit.');
```

Baris ini mendapatkan ID barang yang akan ditambahkan ke favorit dari rute permintaan menggunakan \$request->route('id'). ID barang ini akan digunakan dalam operasi berikutnya.

```
$barangId = $request->route('id');
```

Baris ini mencari entri favorit yang sudah ada dalam database berdasarkan ID pengguna yang terotentikasi dan ID barang yang diambil sebelumnya. Jika entri favorit sudah ada, maka \$existingFavorite akan berisi entri favorit yang sudah ada, jika tidak ada, maka \$existingFavorite akan bernilai null.

```
$existingFavorite = Favorite::where('user_id', auth()->id())
    ->where('barang_id', $barangId)
    ->first();
```

Baris ini akan dijalankan jika tidak ada entri favorit yang sudah ada dalam database. Maka, akan dibuat entri favorit baru dengan membuat objek Favorite, mengisi propertinya dengan ID pengguna yang terotentikasi dan ID barang yang diambil sebelumnya, dan menyimpannya ke dalam database menggunakan metode save(). Setelah itu, pengguna akan diarahkan kembali ke halaman sebelumnya dengan pesan sukses 'Barang berhasil ditambahkan ke favorit.'.

```
if (!$existingFavorite) {
    $favorite = new Favorite();
    $favorite->user_id = auth()->id();
    $favorite->barang_id = $barangId;
    $favorite->save();

    return redirect()->back()->with('success', 'Barang berhasil ditambahkan ke favorit.');
```

Baris ini akan dijalankan jika entri favorit yang sudah ada ditemukan dalam database. Pengguna akan diarahkan kembali ke halaman sebelumnya dengan pesan error 'Barang sudah ada di favorit.'

```
return redirect()->back()->with('error', 'Barang sudah ada di favorit.');
```

- **Fungsi Menghapus Favorite**

```
public function remove($id)
{
    Favorite::destroy($id);
    return redirect()->back();
}
```

Baris ini menggunakan metode destroy() dari model Favorite untuk menghapus entri favorit yang memiliki ID yang sesuai dengan \$id. Metode destroy() akan menghapus entri favorit berdasarkan nilai kunci utamanya.

```
Favorite::destroy($id);
```

Baris ini mengarahkan pengguna kembali ke halaman sebelumnya setelah proses penghapusan selesai.

```
return redirect()->back();
```

- **Source Code View Favorite**

```
@extends('favorite.layouts.template')

@section('favorite')
<!-- ===== Portfolio Section ===== -->
<section id="portfolio" class="portfolio">
    <div class="container">
        <div class="section-title">
            <h2>FAVORITE</h2>
        </div>
        <div class="row portfolio-container">
            @foreach ($favorites as $favorite)
                <div class="col-lg-4 col-md-6 portfolio-item filter-app">
                    <div class="portfolio-wrap">
                        <!-- Gambar -->
                        
                        <div class="portfolio-info">
                            <!-- Merek -->
                            <h4>{{ $favorite->item->merek }}</h4>
                            <!-- Harga -->
                            <p>{{ $favorite->item->harga }}</p>
                            <div class="portfolio-links">
                                <!-- Tombol untuk menghapus dari favorit -->
                                <a href="{{ route('remove.favorite', $favorite->id) }}"
                                    title="Hapus Dari Favorite"><i class="bx bx-minus"></i></a>
                                <!-- Tombol untuk redirect ke detail -->
                                <a href="{{ route('detail.show', $favorite->item->id_barang) }}"
                                    title="More Details"><i class="bx bx-link"></i></a>
                            </div>
                        </div>
                    </div>
                </div>
            @endforeach
        </div>
    </section>
<!-- End Portfolio Section -->
@endsection
```

- **Tampilan Favorite**

ADDICTIVE CLOTHING

FAVORITE



Addictive Clothing

Jln.Tosaren 1,
Kelurahan Tosaren,
Kecamatan Pesantren Kota Kediri
RT02/RW01

No Handphone: +62857-8425-8613
Email: addictive_clothing@example.com

Useful Links

- > Beranda
- > Layanan
- > Produk
- > Tentang Kami

Our Services

- > Kaos lusinan
- > Kaos Jearsty
- > Sablon DTF
- > Sablon DTG
- > Sablon Plastisol

Our Social Networks

Official Merchandise



© Copyright Addictive Clothing. All Rights Reserved



4.1.6 Checkout

- Fungsi Menampilkan Halaman Checkout

```
public function checkout($id_barang){  
    $barang = Barang::find($id_barang);  
    $checkout = Pesanan::where('barang_id', $id_barang)->first();  
    return view('checkout.checkout', compact('barang', 'checkout'));  
}
```

Baris ini mengambil data barang dari tabel barang berdasarkan ID barang yang diberikan menggunakan metode find(). Data barang ini kemudian akan disimpan dalam variabel \$barang.

```
$barang = Barang::find($id_barang);
```

Baris ini mencari data pesanan yang terkait dengan ID barang yang diberikan dalam tabel pesanan menggunakan metode where('barang_id', \$id_barang)->first(). Data pesanan ini akan disimpan dalam variabel \$checkout. Jika tidak ada data pesanan yang terkait, nilai \$checkout akan berupa null.

```
$checkout = Pesanan::where('barang_id', $id_barang)->first();
```

Baris ini mengembalikan tampilan (view) dengan nama 'checkout.checkout'. Tampilan ini akan diisi dengan data yang dikirimkan melalui parameter compact('barang', 'checkout'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$barang dan \$checkout akan tersedia di tampilan dengan nama yang sama.

```
return view('checkout.checkout', compact('barang', 'checkout'));
```


- **Fungsi Proses Checkout**

```
public function process(CheckoutRequest $request){
    $user = Auth::user();
    $barangId = $request->input('barang_id');
    $alamat = $request->input('alamat');
    $provinsi = $request->input('provinsi');
    $kabupaten = $request->input('kabupaten');
    $kodePos = $request->input('kode_pos');
    $nomorHp = $request->input('nomor_hp');
    $metodePembayaran = $request->input('pembayaran');

    $checkout = new Pesanan();
    $checkout->user_id = $user->id;
    $checkout->barang_id = $barangId;
    $checkout->alamat = $alamat;
    $checkout->provinsi = $provinsi;
    $checkout->kabupaten = $kabupaten;
    $checkout->kode_pos = $kodePos;
    $checkout->nomor_hp = $nomorHp;
    $checkout->metode_pembayaran = $metodePembayaran;
    $checkout->save();

    return Redirect::route('pesanan')->with('success', 'Checkout berhasil');
}
```

Baris ini mengambil objek pengguna yang sedang terotentikasi saat ini menggunakan `Auth::user()`. Objek pengguna ini akan digunakan untuk mendapatkan ID pengguna yang akan digunakan dalam proses checkout.

```
$user = Auth::user();
```

Baris ini mengambil nilai dari input form checkout menggunakan metode `input()` pada objek `$request`. Nilai-nilai ini akan disimpan dalam variabel yang sesuai untuk digunakan dalam proses checkout selanjutnya.

```
$barangId = $request->input('barang_id');
$alamat = $request->input('alamat');
$provinsi = $request->input('provinsi');
$kabupaten = $request->input('kabupaten');
$kodePos = $request->input('kode_pos');
$nomorHp = $request->input('nomor_hp');
$metodePembayaran = $request->input('pembayaran');
```

Baris ini membuat objek `Pesanan` baru dan mengisi propertinya dengan nilai-nilai yang telah diambil sebelumnya. Properti `user_id` diisi dengan ID pengguna yang terotentikasi, properti `barang_id` diisi dengan ID barang yang diambil dari input form checkout, dan properti lainnya diisi dengan nilai-nilai yang sesuai dari input form. Setelah itu, objek `Pesanan` disimpan ke dalam database menggunakan metode `save()`.

```
$checkout = new Pesanan();
$checkout->user_id = $user->id;
$checkout->barang_id = $barangId;
$checkout->alamat = $alamat;
$checkout->provinsi = $provinsi;
$checkout->kabupaten = $kabupaten;
$checkout->kode_pos = $kodePos;
$checkout->nomor_hp = $nomorHp;
$checkout->metode_pembayaran = $metodePembayaran;
$checkout->save();
```

Baris ini mengarahkan pengguna ke rute dengan nama 'pesanan' setelah proses checkout selesai. Pengguna juga akan membawa pesan sukses 'Checkout berhasil' yang akan ditampilkan.

```
return Redirect::route('pesanan')->with('success', 'Checkout berhasil');
```

- **Source Code View**

```


@extends('checkout.layouts.base')
@section('checkout')
    <?php
    use Collective\Html\FormFacade;
    use Collective\Html\HtmlFacade;
    ?>
    <main id="main">
        <section id="breadcrumbs" class="breadcrumbs">
            <div class="container">
                <div class="d-flex justify-content-between align-items-center">
                    <h2>CHECKOUT</h2>
                </div>
            </div>
        </section>
        <div class="container">
            <!-- ===== Breadcrumbs ===== -->
            <div class="box-checkout">
                <div class="column-checkout">
                    @if ($barang)
                        <div class="img-produk">
                            
                        </div>
                        <div class="row-checkout">
                            <h4>{{ $barang->merek }}</h4>
                            <h6>{{ nl2br(e($barang->deskripsi)) }}</h6>
                            <h6>Rp. {{ $barang->harga }}</h6>
                            {!! FormFacade::open(['route' => 'checkout.add']) !!}
                            {!! FormFacade::hidden('barang_id', $barang->id_barang) !!}
                            <label class="w-100 mb-3">
                                Nomor HP <br>
                                {!! FormFacade::number('nomor_hp', null, ['class' => 'form-control']) !!}
                            </label>
                            <label class="w-100 mb-3">
                                Alamat <br>
                                {!! FormFacade::text('alamat', null, ['class' => 'form-control']) !!}
                            </label>
                            <label class="w-100 mb-3">
                                Provinsi <br>
                                {!! FormFacade::text('provinsi', null, ['class' => 'form-control']) !!}
                            </label>
                            <label class="w-100 mb-3">
                                Kabupaten / Kota <br>
                                {!! FormFacade::text('kabupaten', null, ['class' => 'form-control']) !!}
                            </label>
                            <label class="w-100 mb-3">
                                Kode POS <br>
                                {!! FormFacade::number('kode_pos', null, ['class' => 'form-control']) !!}
                            </label>
                            <h3 class="text-judul mt-5">Metode Pembayaran</h3>
                            <!-- <label class="w-100 mb-3 rounded border p-2">
                                {!! FormFacade::radio('pembayaran', 'Transfer Bank') !!}
                                Transfer Bank
                            </label> -->
                            <label class="w-100 mb-3 rounded border p-2">
                                {!! FormFacade::radio('pembayaran', 'Cash on Delivery (COD)') !!}
                                Cash on Delivery (COD)
                            </label>
                        <!-- KOLOM 2 -->
                        <div class="card">
                            <div class="card-header bg-white">
                                <h3 class="text-judul">Detail Pembayaran</h3>
                            </div>
                            <div class="card-body">
                                <!-- <div class="row mt-2 mb-2">
                                    <div class="col-md"><small>Biaya pengiriman</small></div>
                                    <div class="col-md">Rp. {{ $checkout->ongkir }}</div>
                                </div> -->
                                <div class="row mt-2 mb-2">
                                    <div class="col-md"><small>Harga</small></div>
                                    <div class="col-md">Rp. {{ $barang->harga }}</div>
                                </div>
                                <!-- Total -->
                                <!-- @if ($barang && $checkout)
                                    <div class="row mt-2 mb-2">
                                        <div class="col-md"><small>Total</small></div>
                                        <div class="col-md">Rp. {{ number_format($total, 0, ',', '.') }}</div>
                                    </div>
                                </if>
                                {!! FormFacade::hidden('total', $total) !!}
                            </div>
                            @endif
                        </div>
                        <div class="card-footer">
                            {!! FormFacade::submit('CHECKOUT', ['class' => 'btn btn-success w-100']) !!}
                        </div>
                    </div>
                </div>
                {!! FormFacade::close() !!}
            </div>
        @else
            <p>Barang tidak ditemukan.</p>
        @endif
    </div>
</main>
@endsection

```

- **Tampilan Checkout**

ADDICTIVE CLOTHING

CHECKOUT



T-Shirt HYPE ASTRO // Addict Clothing - S

PRE ORDER - HYPEASTRO• Black Colour• Local Size• Cotton Combed 30s• DTF• Cash / Atm
BCA• Bebas kirim daerah Kediri (Luar Kediri bisa kirim via JNE/JNT)

Rp. 95.000

Nomor HP

Alamat

Provinsi

Kabupaten / Kota

Kode POS

Metode Pembayaran

☐ Cash on Delivery (COD)

Detail Pembayaran

Harga	Rp. 95.000
-------	------------

CHECKOUT

4.1.7 Pesanan Anda

- **Fungsi Menampilkan Pesanan anda**

```
public function index(){
    $user_id = Auth::id();
    $pesanan = Pesanan::with('barang')->where('user_id', $user_id)->get();

    return view('pesanan.pesanan', compact('pesanan'));
}
```

Baris ini mengambil ID pengguna yang sedang terotentikasi saat ini menggunakan `Auth::id()`. ID pengguna ini akan digunakan dalam query untuk mengambil pesanan terkait dengan pengguna tersebut.

```
$user_id = Auth::id();
```

Baris ini mengambil data pesanan dari tabel pesanan yang terkait dengan ID pengguna yang diberikan. Data pesanan ini juga akan memuat data barang terkait dengan menggunakan metode `with('barang')`. Metode `get()` digunakan untuk mengambil semua hasil query. Hasil query ini akan disimpan dalam variabel `$pesanan`.

```
$pesanan = Pesanan::with('barang')->where('user_id', $user_id)->get();
```

Baris ini mengembalikan tampilan (view) dengan nama `'pesanan.pesanan'`. Tampilan ini akan diisi dengan data pesanan yang dikirimkan melalui parameter `compact('pesanan')`. Fungsi `compact()` digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel `$pesanan` akan tersedia di tampilan dengan nama yang sama.

```
return view('pesanan.pesanan', compact('pesanan'));
```

- **Source Code View**

```
@extends('pengelola.layouts.template')
@section('konfirmasi')
<section id="breadcrumbs" class="breadcrumbs">
  <div class="container">

    <div class="d-flex justify-content-between align-items-center">
      <h2>Konfirmasi Status Pesanan</h2>
    </div>

  </div>
</section>
<div class="container-user">
  <div class="item">
    <table class="table-bordered table">
      <thead>
        <tr>
          <th>No</th>
          <th>Nama</th>
          <th>Email</th>
          <th>Metode</th>
          <th>Status</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        @foreach ($users as $user)
          <tr>
            <td>{{ $user->id }}</td>
            <td>{{ $user->user->name }}</td>
            <td>{{ $user->user->email }}</td>
            <td>{{ $user->metode_pembayaran }}</td>
            <td>{{ $user->status }}</td>
            <td><a href="{{ route('konfirmasi-user', ['id' => $user->id]) }}" class="
            btn btn-warning">Konfirmasi</a></td>
          </tr>
        @endforeach
      </tbody>
    </table>
  </div>
</div>
@endsection
```

- Tampilan Pesanan Anda

ADDICTIVE CLOTHING

Pesanan Anda

Merek	Harga	Alamat	Status
T Shirt Addict Clothing V.3 - S	135.000	Sidoarjo	menunggu pembayaran

4.1.8 Profile

- Fungsi Menampilkan Halaman Profile

```
public function index(){
    $user = User::findOrFail(Auth::id());

    return view('profile.profile', compact('user'));
}
```


Baris ini mengambil objek pengguna yang sedang terotentikasi saat ini dengan menggunakan `Auth::id()` untuk mendapatkan ID pengguna yang terotentikasi, kemudian menggunakan `findOrFail()` untuk mencari pengguna dengan ID tersebut dalam tabel `users`. Jika pengguna tidak ditemukan, maka akan menghasilkan sebuah exception.

```
$user = User::findOrFail(Auth::id());
```

Baris ini mengembalikan tampilan (view) dengan nama `'profile.profile'`. Tampilan ini akan diisi dengan data pengguna yang dikirimkan melalui parameter `compact('user')`. Fungsi `compact()` digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel `$user` akan tersedia di tampilan dengan nama yang sama.

```
return view('profile.profile', compact('user'));
```

- **Source Code Profile**



Admin

Ubah Profil

Informasi Kontak

Alamat :
Masih Belum Ada

Alamat Email
admin@gmail.com

Nomor Telepon
Masih Belum Ada

Addictive Clothing

Jln. Tosaren 1,
Kelurahan Tosaren,
Kecamatan Pesantren Kota Kediri
RT02/RW01

Phone: +62 888 8888 8888
Email: addictive_clothing@example.com

Useful Links

Beranda

Layanan

Produk

Tentang Kami

Our Services

Kaos lusinan

Kaos Jearsy

Sablon DTF

Sablon DTG

Sablon Plastisol

Our Social Networks

Official Merchandise

Instagram WhatsApp Facebook

4.1.9 Ubah Profile

- Fungsi Menampilkan Halaman Ubah Profile

```
public function ubahshow(){
    $user = User::findOrFail(Auth::id());
    return view('profile.ubahprofil', compact('user'));
}
```

Baris ini mengambil objek pengguna yang sedang terotentikasi saat ini dengan menggunakan `Auth::id()` untuk mendapatkan ID pengguna yang terotentikasi, kemudian menggunakan `findOrFail()` untuk mencari pengguna dengan ID tersebut dalam tabel users. Jika pengguna tidak ditemukan, maka akan menghasilkan sebuah exception.

```
$user = User::findOrFail(Auth::id());
```

Baris ini mengembalikan tampilan (view) dengan nama 'profile.ubahprofil'. Tampilan ini akan diisi dengan data pengguna yang dikirimkan melalui parameter `compact('user')`. Fungsi `compact()` digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel `$user` akan tersedia di tampilan dengan nama yang sama.

```
return view('profile.ubahprofil', compact('user'));
```

- Fungsi Ubah Profile


```

public function ubah(Request $request){
    $user = User::findOrFail(Auth::id());

    $validatedData = $request->validate([
        'address' => 'required',
        'email' => 'required|email',
        'phone_number' => 'required',
    ]);

    $user->address = $request->input('address');
    $user->email = $request->input('email');
    $user->phone_number = $request->input('phone_number');
    $user->save();

    return redirect()->route('profile')->with('success', 'Profil berhasil diubah!');
}

```

Baris ini mengambil objek pengguna yang sedang terotentikasi saat ini dengan menggunakan `Auth::id()` untuk mendapatkan ID pengguna yang terotentikasi, kemudian menggunakan `findOrFail()` untuk mencari pengguna dengan ID tersebut dalam tabel users. Jika pengguna tidak ditemukan, maka akan menghasilkan sebuah exception.

```

$user = User::findOrFail(Auth::id());

```

Baris ini melakukan validasi data yang diterima dari permintaan `$request`. Data ini akan divalidasi dengan aturan-aturan yang diberikan. Dalam hal ini, address harus diisi, email harus diisi dan memiliki format yang valid, dan phone_number harus diisi.

```

$validatedData = $request->validate([
    'address' => 'required',
    'email' => 'required|email',
    'phone_number' => 'required',
]);

```

Baris ini mengambil nilai dari input form yang diterima melalui `$request` menggunakan metode `input()` dan mengubah properti address, email, dan phone_number pada objek `$user` dengan nilai-nilai tersebut. Setelah itu, perubahan data profil pengguna disimpan ke dalam database menggunakan metode `save()`.

```

$user->address = $request->input('address');
$user->email = $request->input('email');
$user->phone_number = $request->input('phone_number');
$user->save();

```

Baris ini mengarahkan pengguna ke rute dengan nama 'profile' setelah proses perubahan data profil selesai. Pengguna juga akan membawa pesan sukses 'Profil berhasil diubah!' yang akan ditampilkan.

```

return redirect()->route('profile')->with('success', 'Profil berhasil diubah!');

```

- Source Code View

```
@extends('profile.layouts.base')
@section('ubah')
    <section id="breadcrumbs" class="breadcrumbs">
        <div class="container">


            <div class="d-flex justify-content-between align-items-center">
                <h2>Ubah Profil</h2>
            </div>

        </div>
    </section>
    <div class="container py-5">
        {!! Form::open(['route' => 'profile.ubah']) !!}
        <div class="row">
            <div class="col-lg-4">
                <div class="card p-5 text-center">
                    <div class="card-body">
                        
                        <h2>{{ $user->name }}</h2>
                        {{-- Button Submit --}}
                        {!! Form::submit('Ubah Profile', ['class' => 'btn btn-primary btn-sm']) !!}
                    </div>
                </div>
            </div>
            <div class="col-lg-8">
                <div class="mb-5 rounded border p-5 shadow">
                    <h2>Informasi Kontak</h2>
                    <div class="row">
                        <div class="col-lg-6">
                            {{-- Form Input --}}
                            <p class="card-text">
                                <span class="text-muted d-block mb-1">Alamat :</span>
                                <i class="fa-solid fa-map me-2 text-success"></i>
                                {!! Form::text('address', $user->address, ['class' => 'form-control']) !!}
                            </p>
                            <p class="card-text">
                                <span class="text-muted d-block mb-1">Alamat Email</span>
                                <i class="fa-solid fa-envelope me-2 text-success"></i>
                                {!! Form::text('email', $user->email, ['class' => 'form-control']) !!}
                            </p>
                            <p class="card-text">
                                <span class="text-muted d-block mb-1">Nomor Telepon</span>
                                <i class="fa-solid fa-phone me-2 text-success"></i>
                                {!! Form::text('phone_number', $user->phone_number, ['class' => 'form-control']) !!}
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        {!! Form::close() !!}
    </div>
</div>
@endsection
```

- Tampilan Ubah Profile

ADDICTIVE CLOTHING

Ubah Profil



Admin

[Ubah Profile](#)

Informasi Kontak

Alamat :

Alamat Email

Nomor Telepon

Addictive Clothing

Jln. Tosaren 1,
Kelurahan Tosaren,
Kecamatan Pesantren Kota Kediri
RT02/RW01

Phone: +62 888 8888 8888
Email: addictive_clothing@example.com

Useful Links

[Beranda](#)
[Layanan](#)
[Produk](#)
[Tentang Kami](#)

Our Services

[Kaos lusinan](#)
[Kaos Jearsty](#)
[Sablon DTF](#)
[Sablon DTG](#)
[Sablon Plastisol](#)

Our Social Networks

Official Merchandise



© Copyright Addictive Clothing. All Rights Reserved

4.1.10 Admin Input Barang

- **Fungsi Menampilkan Halaman Input**

```
public function index()
{
    return view('pengelola.input');
}
```

Baris itu menampilkan file view yang berlokasi di direktori resources/views/pengelola/input.

- **Fungsi Menambahkan Barang**

```

public function store(Request $request){
    $validatedData = $request->validate([
        'merek' => 'required',
        'gambar' => 'required',
        'harga' => 'required|numeric',
        'deskripsi' => 'required',
    ]);

    $barang = new Barang;
    $barang->merek = $request->merek;
    $barang->gambar = $request->gambar;
    $barang->harga = $request->harga;
    $barang->deskripsi = $request->deskripsi;

    $barang->save();

    return redirect()->route('input.success');
}

```

Baris ini melakukan validasi data yang diterima dari permintaan \$request. Data ini akan divalidasi dengan aturan-aturan yang diberikan. Dalam hal ini, merek, gambar, harga, dan deskripsi harus diisi. Selain itu, harga harus berupa angka (numeric).

```

$validatedData = $request->validate([
    'merek' => 'required',
    'gambar' => 'required',
    'harga' => 'required|numeric',
    'deskripsi' => 'required',
]);

```

Baris ini membuat objek baru dari model Barang dan mengisi properti-propertinya dengan nilai-nilai yang diterima dari \$request. Setelah itu, objek barang tersebut disimpan ke dalam database menggunakan metode save().

```

$barang = new Barang;
$barang->merek = $request->merek;
$barang->gambar = $request->gambar;
$barang->harga = $request->harga;
$barang->deskripsi = $request->deskripsi;

$barang->save();

```

```

return redirect()->route('input.success');

```

Baris ini mengarahkan pengguna ke rute dengan nama 'input.success' setelah proses penyimpanan data barang selesai. Pengguna akan diarahkan ke halaman yang menunjukkan bahwa proses input berhasil.

- **Source Code View**

```

@extends('pengelola.layouts.template')
@section('input')
    <section id="breadcrumbs" class="breadcrumbs">
        <div class="container">

            <div class="d-flex justify-content-between align-items-center">
                <h2>Input Barang</h2>
            </div>

        </div>
    </section>
    <section class="form">
        <div class="form-container">
            <form action="{{ route('input.store') }}" method="POST">
                @csrf
                <div class="mb-3">
                    <label for="merek" class="form-label">Merek</label>
                    <input type="text" class="form-control" id="merek" name="merek">
                </div>
                <div class="mb-3">
                    <label for="merek" class="form-label">Gambar</label>
                    <input type="text" class="form-control" id="gambar" name="gambar">
                </div>
                <div class="mb-3">
                    <label for="merek" class="form-label">Harga</label>
                    <input type="number" class="form-control" id="harga" name="harga">
                </div>
                <div class="mb-3">
                    <label for="merek" class="form-label">Deskripsi</label>
                    <input type="text" class="form-control" id="deskripsi" name="deskripsi">
                </div>

                <button type="submit" class="btn btn-success">Submit</button>
            </form>
        </div>
    </section>
@endsection

```

- **Tampilan Input Barang**

ADDITIVECLOTHING ADMIN

INPUT BARANG USER KONFIRMASI PEMBAYARAN LOGOUT

Input Barang

Merek

Gambar

Harga

Deskripsi

Submit

4.1.11 Admin Barang

- **Fungsi Menampilkan Halaman Barang**

```
public function index(){
    $barangs = Barang::all();
    return view('pengelola.barang', compact('barangs'));
}
```

Baris ini mengambil semua data barang dari tabel barang dengan menggunakan model Barang dan metode all(). Data barang tersebut akan disimpan dalam variabel \$barangs.

```
$barangs = Barang::all();
```

Baris ini mengembalikan tampilan (view) dengan nama 'pengelola.barang'. Tampilan ini akan diisi dengan data barang yang dikirimkan melalui parameter compact('barangs'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$barangs akan tersedia di tampilan dengan nama yang sama.

```
return view('pengelola.barang', compact('barangs'));
```

- **Fungsi Hapus Barang**

```
public function destroy($id_barang){
    $barang = Barang::findOrFail($id_barang);
    $barang->delete();

    return redirect()->route('barang')->with('success', 'Data has been deleted successfully!');
}
```

Baris ini mencari data barang berdasarkan ID barang yang diterima dari parameter menggunakan metode findOrFail(). Jika barang tidak ditemukan, maka akan menghasilkan sebuah exception.

```
$barang = Barang::findOrFail($id_barang);
```

Baris ini menghapus data barang dengan menggunakan metode delete() pada objek barang yang ditemukan sebelumnya.

```
$barang->delete();
```

Baris ini mengarahkan pengguna ke rute dengan nama 'barang' setelah proses penghapusan data barang selesai. Pengguna juga akan membawa pesan sukses 'Data has been deleted successfully!' yang akan ditampilkan.

```
return redirect()->route('barang')->with('success', 'Data has been deleted successfully!');
```

- **Source Code View**

```

@extends('pengelola.layouts.template')
@section('barang')
    <section id="breadcrumbs" class="breadcrumbs">
        <div class="container">

            <div class="d-flex justify-content-between align-items-center">
                <h2>Barang</h2>
            </div>

        </div>
    </section>
    <div class="container-barang">
        <table class="table-bordered table">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Merek</th>
                    <th>Gambar</th>
                    <th>Harga</th>
                    <th>Deskripsi</th>
                    <th>Created At</th>
                    <th>Update At</th>
                    <th>Aksi</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($barang as $barang)
                    <tr>
                        <td>{{ $barang->id_barang }}</td>
                        <td>{{ $barang->merek }}</td>
                        <td>{{ $barang->gambar }}</td>
                        <td>{{ $barang->harga }}</td>
                        <td>{{ $barang->deskripsi }}</td>
                        <td>{{ $barang->created_at }}</td>
                        <td>{{ $barang->updated_at }}</td>
                        <td>
                            <div class="btn-edit">
                                <a href="{{ route('barang.edit', $barang->id_barang) }}" class="
"btn btn-warning">Edit</a>
                            </div>
                            <div class="btn-hapus">
                                <form action="{{ route('barang.delete', $barang->id_barang) }}" method="POST"
>
                                    @csrf
                                    @method('DELETE')
                                    <button type="submit"
                                        onclick="return confirm('Apakah Anda yakin ingin menghapus data ini?')
">
                                        class="btn btn-danger">Hapus</button>
                                </form>
                            </div>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
@endsection

```

- **Tampilan Admin Barang**

Barang

No	Merek	Gambar	Harga	Deskripsi	Created At	Update At	Aksi
1	T-shirt streetwear // Kitsune Myth - S	Kitsune_Myth_S.png	135.000	Vol #1PRE ORDER - KITSUNE MYTH • Black Colour • Local Size • Catton Combad 30s • DTF / DTG • Cash / Atm BCA • Bebas kirim daerah Kediri (Luar Kediri bisa kirim via JNE/JNT)		2023-06-10 14:11:39	Edit Hapus
2	T-Shirt HYPE ASTRO // Addict Clothing - S	Addict_Clothing_S.png	95.000	PRE ORDER - HYPE ASTRO• Black Colour• Local Size• Catton Combad 30s• DTF• Cash / Atm BCA• Bebas kirim daerah Kediri (Luar Kediri bisa kirim via JNE/JNT)		2023-06-09 09:28:43	Edit Hapus
3	T Shirt Addict Clothing V.3 - S	Addict_Clothing_V.3_S.png	135.000	Vol #3PRE ORDER - ADDICT CLOTHING• Black Colour• Local Size• Catton Combad 30s• DTF / DTG• Cash / Atm BCA• Bebas kirim daerah Kediri (Luar Kediri bisa kirim via JNE/JNT)		2023-06-09 09:28:46	Edit Hapus
4	T-shirt Streetwear // Astronot Desain - M	Astronot_Desain_M.png	90.000	Vol #2PRE ORDER - ASTRONAUTPesanan dahulu barang ready setelah 3-4 hari pemesanan.Keterangan :• Black Colour• Local Size S - XXL• Catton Combad 30s• Plastisol• Bebas kirim daerah Kediri (Luar Kediri bisa kirim via JNE/JNT)		2023-06-09 09:28:49	Edit Hapus



4.1.12 Edit Barang

- **Fungsi Menampilkan Halaman Edit Barang**

```
public function edit($id_barang){
    $barang = Barang::findOrFail($id_barang);
    return view('pengelola.edit', compact('barang'));
}
```

Baris ini mencari data barang berdasarkan ID barang yang diterima dari parameter menggunakan metode `findOrFail()`. Jika barang tidak ditemukan, maka akan menghasilkan sebuah exception.

```
$barang = Barang::findOrFail($id_barang);
```

Baris ini mengembalikan tampilan (view) dengan nama 'pengelola.edit'. Tampilan ini akan diisi dengan data barang yang dikirimkan melalui parameter `compact('barang')`. Fungsi `compact()` digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel `$barang` akan tersedia di tampilan dengan nama yang sama.

```
return view('pengelola.edit', compact('barang'));
```

- **Fungsi Edit Barang**

```
public function update(Request $request, $id_barang){
    $barang = Barang::findOrFail($id_barang);
    $barang->merek = $request->input('merek');
    $barang->gambar = $request->input('gambar');
    $barang->harga = $request->input('harga');
    $barang->deskripsi = $request->input('deskripsi');
    $barang->save();

    return redirect()->route('barang')->with('success', 'Data has been updated successfully!');
}
```

Baris ini mencari data barang berdasarkan ID barang yang diterima dari parameter menggunakan metode `findOrFail()`. Jika barang tidak ditemukan, maka akan menghasilkan sebuah exception.


```
$barang = Barang::findOrFail($id_barang);
```

Baris-baris ini mengambil nilai-nilai baru dari formulir yang dikirimkan melalui objek Request dan memperbarui nilai-nilai tersebut pada objek barang yang ditemukan sebelumnya. Nilai-nilai tersebut diakses menggunakan metode `input()` dengan menyediakan nama kolom pada formulir.

```
$barang->merek = $request->input('merek');  
$barang->gambar = $request->input('gambar');  
$barang->harga = $request->input('harga');  
$barang->deskripsi = $request->input('deskripsi');  
$barang->save();
```

Baris ini mengarahkan pengguna ke rute dengan nama 'barang' setelah proses pembaruan data barang selesai. Pengguna juga akan membawa pesan sukses 'Data has been updated successfully!' yang akan ditampilkan.

```
return redirect()->route('barang')->with('success', 'Data has been updated successfully!');
```

- **Source Code View**

```
@extends('pengelola.layouts.template')  
@section('edit')  
    <section id="breadcrumbs" class="breadcrumbs">  
        <div class="container">  
  
            <div class="d-flex justify-content-between align-items-center">  
                <h2>Edit Barang</h2>  
            </div>  
  
        </div>  
    </section>  
    <div class="container">  
        <div class="item">  
            <form action="{{ route('barang.update', $barang->id_barang) }}" method="POST">  
                @csrf  
                @method('PUT')  
                <div class="mb-3">  
                    <label for="merek" class="form-label">Merek</label>  
                    <input type="text" class="form-control" name="merek" value="{{ $barang->merek }}" required>  
                </div>  
                <div class="mb-3">  
                    <label for="gambar" class="form-label">Gambar</label>  
                    <input type="text" class="form-control" name="gambar" value="{{ $barang->gambar }}" required>  
                </div>  
                <div class="mb-3">  
                    <label for="harga" class="form-label">Harga</label>  
                    <input type="number" class="form-control" name="harga" value="{{ $barang->harga }}" required>  
                </div>  
                <div class="mb-3">  
                    <label for="deskripsi" class="form-label">Deskripsi</label>  
                    <input type="text" class="form-control" name="deskripsi" value="{{ $barang->deskripsi }}" required>  
                </div>  
                <input type="submit" class="btn btn-warning" value="Update">  
            </form>  
        </div>  
    </div>  
@endsection
```

- **Tampilan Edit Barang**

Edit Barang

Merek

Gambar

Harga

Deskripsi

4.1.13 Admin User

- Fungsi Menampilkan Halaman User

```
public function index(){  
    $users = UserTable::all();  
    return view('pengelola.user', compact('users'));  
}
```

Baris ini mengambil semua data dari tabel UserTable menggunakan metode all() dan menyimpannya ke dalam variabel \$users.

```
$users = UserTable::all();
```

Baris ini mengembalikan tampilan (view) dengan nama 'pengelola.user'. Tampilan ini akan diisi dengan data pengguna yang dikirimkan melalui parameter compact('users'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$users akan tersedia di tampilan dengan nama yang sama.

```
return view('pengelola.user', compact('users'));
```

- Source Code View

```

@extends('pengelola.layouts.template')
@section('user')
<section id="breadcrumbs" class="breadcrumbs">
    <div class="container">

        <div class="d-flex justify-content-between align-items-center">
            <h2>User</h2>
        </div>

    </div>
</section>
<div class="container-user">
    <div class="item">
        <table class="table-bordered table">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Nama</th>
                    <th>Email</th>
                    <th>Nomor HP</th>
                    <th>Alamat</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($users as $user)
                    <tr>
                        <td>{{ $user->id }}</td>
                        <td>{{ $user->name }}</td>
                        <td>{{ $user->email }}</td>
                        <td>{{ $user->phone_number }}</td>
                        <td>{{ $user->address }}</td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</div>
@endsection

```

• Tampilan User

ADDITIVECLOTHING ADMIN

INPUT BARANG USER KONFIRMASI PEMBAYARAN LOGOUT

User

No	Nama	Email	Nomor HP	Alamat
3	aaaaa	aaaaa@gmail.com		
4	coba1	coba1@gmail.com		
5	QWERTY	qwerty1234@gmail.com		
6	Bilal Shandy	bilalshandy@gmail.com		
8	Bilal Shandyarta	44006bilal@gmail.com		
9	Bilal Shandy	bilalshandy04@gmail.com		
10	coba	44006piyu@gmail.com		
11	Akun Percobaan	akuncoba@gmail.com	098765432123	Wonoayu
12	AKUN COBA 2	akuncoba2@gmail.com		
14	CobaAPI	bilalapi@gmail.com		sidokare
15	CobaAPI2	bilalapi2@gmail.com		sidokare
16	CobaAPI3	bilalapi3@gmail.com		sidokare
17	CobaAPI4	bilalapi4@gmail.com	098765432123	sidokare
18	dev	dev@gamil.com		
19	Admin	admin@gmail.com		
20	LaMi	lami@gmail.com		



4.1.14 Admin Pesanan

- Fungsi Menampilkan Halaman Pesanan

```
public function index(){  
    $users = Pesanan::all();  
    return view('pengelola.konfirmasi', compact('users'));  
}
```

Baris ini mengambil semua data dari model Pesanan menggunakan metode all() dan menyimpannya ke dalam variabel \$users. Model Pesanan diasumsikan sebagai model yang terhubung dengan tabel yang berisi data pesanan.

```
$users = Pesanan::all();
```

Baris ini mengembalikan tampilan (view) dengan nama 'pengelola.konfirmasi'. Tampilan ini akan diisi dengan data pesanan yang dikirimkan melalui parameter compact('users'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$users akan tersedia di tampilan dengan nama yang sama.

```
return view('pengelola.konfirmasi', compact('users'));
```

- Source Code View

```
@extends('pengelola.layouts.template')  
@section('konfirmasi')  
    <section id="breadcrumbs" class="breadcrumbs">  
        <div class="container">  
  
            <div class="d-flex justify-content-between align-items-center">  
                <h2>Konfirmasi Status Pesanan</h2>  
            </div>  
  
        </div>  
    </section>  
    <div class="container-user">  
        <div class="item">  
            <table class="table-bordered table">  
                <thead>  
                    <tr>  
                        <th>No</th>  
                        <th>Nama</th>  
                        <th>Email</th>  
                        <th>Metode</th>  
                        <th>Status</th>  
                        <th>Aksi</th>  
                    </tr>  
                </thead>  
                <tbody>  
                    @foreach ($users as $user)  
                        <tr>  
                            <td>{{ $user->id }}</td>  
                            <td>{{ $user->user->name }}</td>  
                            <td>{{ $user->user->email }}</td>  
                            <td>{{ $user->metode_pembayaran }}</td>  
                            <td>{{ $user->status }}</td>  
                            <td><a href="{{ route('konfirmasi-user', ['id' => $user->id]) }}" class=  
"btn btn-warning">Konfirmasi</a></td>  
                        </tr>  
                    @endforeach  
                </tbody>  
            </table>  
        </div>  
    </div>  
@endsection
```

- Tampilan Admin Pesanan

Konfirmasi Status Pesanan

No	Nama	Email	Metode	Status	Aksi
17	Akun Percobaan	akuncoba@gmail.com	Cash on Delivery (COD)	menunggu pembayaran	Konfirmasi
18	Akun Percobaan	akuncoba@gmail.com	Cash on Delivery (COD)	menunggu pembayaran	Konfirmasi
19	LaMi	lami@gmail.com	Cash on Delivery (COD)	menunggu pembayaran	Konfirmasi
20	Admin	admin@gmail.com	Cash on Delivery (COD)	menunggu pembayaran	Konfirmasi

4.1.15 Konfirmasi Pesanan

- **Fungsi Menampilkan Halaman Konfirmasi Pesanan**

```
public function konfirmasi(Request $request, $id){
    $konfirmasi = Pesanan::findOrFail($id);
    return view('pengelola.konfirmasi-user', compact('konfirmasi'));
}
```

Baris ini mengambil semua data dari model Pesanan menggunakan metode all() dan menyimpannya ke dalam variabel \$users. Model Pesanan diasumsikan sebagai model yang terhubung dengan tabel yang berisi data pesanan.

```
$konfirmasi = Pesanan::findOrFail($id);
```

Baris ini mengembalikan tampilan (view) dengan nama 'pengelola.konfirmasi'. Tampilan ini akan diisi dengan data pesanan yang dikirimkan melalui parameter compact('users'). Fungsi compact() digunakan untuk membuat array dengan kunci yang sama dengan nama variabel dan nilainya dari variabel yang diambil. Dalam hal ini, variabel \$users akan tersedia di tampilan dengan nama yang sama.

```
return view('pengelola.konfirmasi-user', compact('konfirmasi'));
```

- **Fungsi Update Status Pesanan**

```
public function update(Request $request, $id){
    $konfirmasi = Pesanan::findOrFail($id);

    $validatedData = $request->validate([
        'status' => 'required',
    ]);

    $konfirmasi->status = $request->input('status');
    $konfirmasi->save();

    return redirect()->route('konfirmasi')->with('success', 'Status pesanan berhasil diubah!');
}
```

Baris ini mencari data pesanan berdasarkan ID pesanan yang diterima dari parameter menggunakan metode findOrFail(). Jika pesanan tidak ditemukan, maka akan menghasilkan sebuah exception.

```
$konfirmasi = Pesanan::findOrFail($id);
```

Baris ini memvalidasi data yang diterima melalui objek Request. Dalam hal ini, data 'status' harus ada dan bersifat wajib (required).

```
$validatedData = $request->validate([  
    'status' => 'required',  
]);
```

Baris ini mengambil nilai 'status' yang diterima melalui objek Request dan menyimpannya ke dalam properti 'status' dari objek \$konfirmasi. Kemudian, data pesanan yang telah diubah disimpan ke dalam basis data menggunakan metode save().

```
$konfirmasi->status = $request->input('status');  
$konfirmasi->save();
```

Baris ini mengarahkan pengguna ke rute dengan nama 'konfirmasi' setelah mengupdate status pesanan. Selain itu, pesan sukses 'Status pesanan berhasil diubah!' disimpan dalam sesi flash untuk ditampilkan setelah pengalihan halaman.

```
return redirect()->route('konfirmasi')->with('success', 'Status pesanan berhasil diubah!');
```

- **Source Code View**

```
@extends('pengelola.layouts.template')
@section('konfirmasi-user')
    <section id="breadcrumbs" class="breadcrumbs">
        <div class="container">
            <div class="d-flex justify-content-between align-items-center">
                <h2>Konfirmasi Status Pesanan</h2>
            </div>
        </div>
    </section>
    <div class="container-user">
        <div class="item">
            {!! Form::open(['route' => ['konfirmasi-update', $konfirmasi->id], 'method' => 'PUT']) !!}
            <table class="table-bordered table">
                <thead>
                    <tr>
                        <th>No</th>
                        <th>Nama</th>
                        <th>Email</th>
                        <th>Status</th>
                        <th>Aksi</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>{{ $konfirmasi->id }}</td>
                        <td>{{ $konfirmasi->user->name }}</td>
                        <td>{{ $konfirmasi->user->email }}</td>
                        <td>
                            {!! Form::radio('status', 'Menunggu Pembayaran', $konfirmasi->status == 'Menunggu Pembayaran') !!}
                            {!! Form::label('status', 'Menunggu Pembayaran') !!}
                            <br>
                            {!! Form::radio('status', 'Pembayaran Diterima', $konfirmasi->status == 'Pembayaran Diterima') !!}
                            {!! Form::label('status', 'Pembayaran Diterima') !!}
                            <br>
                            {!! Form::radio('status', 'Dibatalkan', $konfirmasi->status == 'Dibatalkan') !!}
                            {!! Form::label('status', 'Dibatalkan') !!}
                        </td>
                        <td>
                            {!! Form::submit('Simpan', ['class' => 'btn btn-primary']) !!}
                        </td>
                    </tr>
                </tbody>
            </table>
            {!! Form::close() !!}
        </div>
    </div>
@endsection
```

• **Tampilan Konfirmasi Pesanan**

ADDICTIVECLOTHINGADMIN

INPUTBARANGUSERKONFIRMASI PEMBAYARANLOGOUT

Konfirmasi Status Pesanan

No	Nama	Email	Status	Aksi
17	Akun Percobaan	akuncoba@gmail.com	<div><input type="radio"/> Menunggu Pembayaran</div> <div><input type="radio"/> Pembayaran Diterima</div> <div><input type="radio"/> Dibatalkan</div>	<div>Simpan</div>

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembuatan yang dilakukan terhadap aplikasi web e-commerce penjualan kaos pada Addictive Clothing, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan adanya perancangan sistem Web E-Commerce Addictive Clothing tentunya akan mempermudah proses transaksi dan customer melalui website yang sudah di bangun.
2. Pemanfaatan website e-commerce sebagai media pemasaran, dapat memudahkan Addictive Clothing memberikan pelayanan terhadap konsumen secara optimal serta dapat diakses kapan saja dan dimana saja.

5.2 Saran

Berdasarkan kesimpulan diatas, maka dapat disimpulkan beberapa saran yang dapat menjadi bahan pertimbangan lebih lanjut dalam upaya peningkatan kualitas system yang telah dibuat. Adapun saran- saran yang ingin disampaikan adalah :

1. Melakukan pembaruan pada *interface* dan sistemnya karena seiring berjalannya waktu akan lahir lagi fitur baru yang lebih memanjakan penggunanya.
2. Dilakukannya maintenance pada system ini, agar penggunanya dapat menggunakan secara berkelanjutan.
3. Web *e-commerce* ini dapat disesuaikan dengan kebutuhan serta perkembangan dari Addictive Clothing pada masa yang akan datang

DAFTAR PUSTAKA

- Abdulloh, R. (2016). Web programing is easy & simple. In *PT Elex Media Komputindo, Jakarta*.
- Andaru, A. (2018). Pengertian database secara umum. In *OSF Prepr. osf.io*.
<https://osf.io/43b5j/download>
- Hastanti, R. P., & Purnama, B. E. (2015). Sistem penjualan berbasis web (e-commerce) pada tata distro kabupaten pacitan. *Bianglala Informatika*.
<https://ejournal.bsi.ac.id/ejurnal/index.php/Bianglala/article/view/581>
- Susandi, D., & Sukisno, S. (2017). Sistem Penjualan Berbasis E-Commerce Menggunakan Metode Objek Oriented pada Distro Dlapak Street Wear. *JSiI (Jurnal Sistem Informasi)*, 4. <https://doi.org/10.30656/jsii.v4i0.368>
- Susanti, D., & Nurdiana, N. (2018). Pengaruh Pemanfaatan Teknologi Informasi Terhadap Kualitas Pelayanan Mahasiswa. *JSiI (Jurnal Sistem Informasi)*. <http://ejurnal.lppmunsera.org/index.php/jsii/article/view/774>
- Yawedani, D., & Wijaya, R. (2017). Pengaruh E-Commerce Terhadap Minat Berwirausaha (Studi Kasus: AMIK Jayanusa Padang). In *Jurnal Resti*.

Link Bootstrap

<https://getbootstrap.com/docs/5.3/>