

FINAL PROJECT

WEB PROGRAMMING

Name:

Aditya Arya Putra

Luthfi Izza Pratama

Bernard Hugo

Garnetio Rafi Datya Nararya

Tharieq Abdurrashid Djalins

NIM:

2201783671

2502034912

2540124450

2201829685

2301933553

Class Program:

School Of Computer Science

Lecturer:

Rani Puspita, S.Kom., M.Kom

People
Innovation
Excellence

UNIVERSITAS BINA NUSANTARA

COMPUTER SCIENCE

2023, ODD SEMESTER

Introduction

Latar Belakang

Landscape retail pada saat ini mengalami transformasi yang signifikan dengan munculnya era digital. Toko-toko tradisional semakin dilengkapi, jika tidak digantikan, oleh platform online. Termasuk salah satunya industri fesyen, khususnya telah menyaksikan perubahan yang mengharuskan pendirian situs web pakaian sebagai komponen yang sangat diperlukan dalam strategi ritel yang sukses.

Konsumen masa kini menunjukkan perubahan besar dalam perilaku pembelian, dengan meningkatnya preferensi belanja online karena kenyamanan, aksesibilitas, dan banyak penawaran produk yang tersedia di internet telah mengubah cara individu berinteraksi dengan pasar ritel. Membangun situs web pakaian menjadi hal yang penting untuk memanfaatkan perubahan preferensi konsumen, menjangkau pengguna lebih luas, dan memenuhi permintaan mereka.

Integrasi fungsi e-commerce dalam situs web pakaian memanfaatkan kemajuan teknologi untuk memudahkan pengalaman berbelanja. Fitur-fitur seperti transaksi online yang aman, rekomendasi yang dipersonalisasi, dan alat uji coba virtual meningkatkan keterlibatan dan kepuasan pelanggan. Mengikuti tren teknologi memastikan situs web pakaian tetap kompetitif dalam dunia digital yang dinamis dan berkembang pesat.

Membangun situs web pakaian bukan sekadar respon terhadap tren ritel kontemporer, namun merupakan strategis bagi bisnis yang ingin berkembang di era digital. Keunggulan jangkauan global, integrasi e-commerce, pengambilan keputusan berdasarkan data, dan visibilitas merek peran penting situs web pakaian yang dirancang dengan baik dan responsif dalam membentuk kesuksesan dan keberlanjutan pengecer mode di pasar modern.

Tujuan

Tujuan pembuatan situs web pakaian tersebut adalah:

1. Dapat dijangkau negara dalam maupun mancanegara
2. Meningkatkan penjualan dan keuntungan
3. Marketing dan promosi
4. Integritas Customer
5. Adaptasi dengan tren-tren terbaru

Batasan

Pada proses pembuatan situs web pakaian banyak kendala yang dialami oleh developer saat progress diantaranya adalah:

1. Target marketing yang spesifik:
2. User Experience
3. Search Engine Optimization
4. Responsive Design
5. Content Management

Manfaat

Manfaat pembentukan situs web pakaian adalah

1. Memberikan pelayanan kemudahan untuk customer
2. Memperluas Pasar
3. Meningkatkan peluang promosi
4. Strategi pemasaran yang efektif
5. Menambah daya tingkat pengikut sosial media

Tools yang digunakan

Berikut adalah tools yang kami gunakan dalam pembuatan web pakaian adalah:

1. Visual Studio Code untuk membuat codingan pada projek laravel.
2. Xampp untuk membuat database yang diperlukan untuk menyimpan data-data.
3. Bahasa pemrograman laravel untuk membuat codingan konten web dan CSS untuk mengatur pengayaan setiap halaman web.

Coding

Route:

```
Route::get('/', [WebController::class, 'home'])->name('home');

Route::get('/home', [WebController::class, 'home'])->name('home');

Route::get('/login', [AuthController::class, 'logIn'])->name('login');

Route::post('validate_login', [AuthController::class, 'validateLogin'])->name('login.validate');

Route::get('/register', [AuthController::class, 'register'])->name('register');

Route::post('validate_register', [AuthController::class,
'validateRegister'])->name('register.validate');

Route::get('/logout', [AuthController::class, 'logOut'])->name('logout');

Route::get('/clothes_detail', [ClothesController::class,
'clothesDetail'])->name('clothes_detail');

Route::get('/add_clothes', [ClothesController::class, 'addClothes'])->name('add_clothes');

Route::post('store_clothes', [ClothesController::class, 'storeClothes'])->name('store_clothes');

Route::get('/search_clothes', [ClothesController::class, 'search'])->name('search_clothes');

Route::get('/cart', [CartController::class, 'cart'])->name('cart');

Route::get('/checkout', [CartController::class, 'checkout'])->name('checkout');

Route::post('/do_checkout', [CartController::class, 'doCheckout'])->name('cart.checkout');

Route::post('add_to_cart', [CartController::class, 'addToCart'])->name('cart.add');

Route::resource('/carts', CartController::class);
```

Berikut, terdapat beberapa routing untuk mengarahkan *user* ke dalam page yang sudah ditentukan. Routing merupakan mekanisme untuk mengarahkan permintaan HTTP ke controller. Setiap route mendefinisikan bagaimana web merespons kepada URL tertentu.

Sebagai contoh, rute **`Route::get('/login', [AuthController::class, 'login']->name('login'))`**; mengindikasikan bahwa ketika pengguna mengakses URL /login dengan metode GET, aplikasi akan mengeksekusi metode login yang terdapat pada AuthController. Rute ini juga diberikan nama login, yang memungkinkan pengembang untuk merujuk rute tersebut dengan mudah di dalam kode lain atau ketika menghasilkan URL ke halaman login. Ini memastikan bahwa navigasi dan alur aplikasi dapat dikelola dengan jelas dan efisien.

Migration:

Create clothes table:

```
<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration {

    public function up(): void {

        Schema::create('clothes', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('image');

            $table->string('type');

            $table->string('brand');

            $table->string('color');

            $table->string('size');

            $table->double('price');

            $table->string('description');

            $table->timestamps();

        });

    }

}
```



```
public function down(): void {  
  
    Schema::dropIfExists('clothes');  
  
}  
  
};
```

Kodingan ini akan membuat sebuah table *food* yang akan menyimpan data-data pakaiannya

Create cart table:

```
<?php  
  
use Illuminate\Database\Migrations\Migration;  
  
use Illuminate\Database\Schema\Blueprint;  
  
use Illuminate\Support\Facades\Schema;  
  
return new class extends Migration{  
  
    public function up(): void {  
  
        Schema::create('cart', function (Blueprint $table) {  
  
            $table->id();  
  
            $table->integer('Quantity');  
  
            $table->integer('Total');  
  
            $table->timestamps();  
  
        });  
  
    }  
  
    public function down(): void {
```

```
Schema::dropIfExists('cart');

}

};
```

Kodingan ini akan membuat tabel *cart* pada *database* dengan kolom-kolom berupa “quantity” yaitu jumlah pakaian dan “total” yaitu total harga. Lalu di database XAMPP phpmyadmin, kolom id diubah menjadi *cart_id* ditambahkan kolom baru bernama “*food_id*” untuk membuat hubungan antara table *cart* dan *food*. Karena dalam halaman *cart* perlu ditunjukkan nama pakaian beserta harganya.

Create checkout table:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {

    public function up(): void {

        Schema::create('checkout', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('phone');

            $table->string('address');

            $table->timestamps();

        });

    }

};
```

```
}  
  
public function down(): void {  
  
    Schema::dropIfExists('checkout');  
  
}  
  
};
```

Kodingan ini akan membuat tabel *checkout* di *database* yang berisi kolom-kolom berupa *name* (nama), *phone* (nomor telepon), dan *address* (alamat rumah).

Model

Clothes:

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Clothes extends Model
{
    use HasFactory;

    protected $table = 'clothes';

    protected $fillable = [
        'name', 'image', 'type', 'brand', 'color', 'size',
        'price', 'description'
    ];

    protected $primaryKey = 'id';

    public function cart(){

        return $this->hasMany(Cart::class, "cart_id", "id");
    }
}
```

```
}  
  
}
```

Kodingan ini akan membuat model bernama “Clothes” yang akan membantu *controller* untuk meminta atribut-atribut berupa: *name* (nama pakaian), *image* (gambar pakaian), *type* (tipe pakaian), *brand* (nama merek pakaian), *color* (warna pakaian), *size* (ukuran pakaian yang tersedia), *price* (harga pakaian), dan *description* (ringkasan mengenai pakaian).

Checkout:

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Checkout extends Model  
{  
    use HasFactory;  
  
    protected $table = 'checkout';  
  
    protected $primaryKey = 'id';  
}
```

```
protected $fillable = [  
    'name', 'phone', 'address'  
];  
}
```

Kodingan ini akan membuat model “Checkout” yang akan membantu *controller* untuk meminta atribut-atribut pada tabel “checkout”. Dimana *primary key* nya adalah id lalu atribut-atributnya berupa *name* (nama pengguna), *phone* (nomor telepon pengguna), dan *address* (alamat pengguna).

Cart:

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Cart extends Model  
{  
    use HasFactory;  
  
    protected $table = 'cart';  
  
    protected $primaryKey = 'cart_id';  
}
```

```
protected $fillable = [
    'food_id', 'user_id', 'quantity'
];

public function clothes(){
    return $this->hasMany(Clothes::class, "id",
"cart_id");
}
}
```

Kodingan ini berisi atribut-atribut dari tabel *cart* untuk membantu *controller*.

Seeder

```
<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
}
```

```
*/  
  
public function run(): void  
{  
    DB::table('clothes')->insert([  
        'name' => 'Blunted Tee',  
        'image' => '1701697742.png',  
        'type' => 'T-shirt',  
        'brand' => 'Cypress Hill',  
        'color' => 'Black',  
        'size' => 'M, L',  
        'price' => 375,  
        'description' => 'Oil based ink screen print,  
crewneck, short sleeved.'  
    ]);  
}  
};
```

Kodingan seeder berisi data-data pakaian yang akan dimasukkan ke tabel clothes di database.

Controller

AuthController:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Hash;
use Session;

use App\Models\User;
use App\Models\Clothes;

use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;

class AuthController extends Controller{

    public function login(){

        return view('login');

    }

    public function register(){

        return view('register');
```

```

}

public function validateRegister(Request $request){

    $request->validate([

        'name'      => 'required|min:5|max:20',

        'email'      => 'required|email|ends_with:@gmail.com|unique:users',

        'password'   => 'required|min:5|max:100',

        'confirm_password' => 'required|min:5|max:100|same:password',

    ]);

    User::create([

        'name' => $request->input('name'),

        'email' => $request->input('email'),

        'password' => Hash::make($request->input('password')),

        'confirm_password' => Hash::make($request->input('confirm_password'))

    ]);

    return redirect('login')->with('success', 'Registration completed');

}

public function validateLogin(Request $request){

    $request->validate([

        'email' => 'required|ends_with:@gmail.com',

        'password' => 'required|min:5|max:255'

    ]);

}

```

```

$credentials = $request->only('email', 'password');

if(Auth::attempt($credentials)){

    return redirect('home');

}

return redirect('login')->with('fail_message', 'Incorrect email or password');

}

public function home(Request $request){

    $clothes = DB::select('select * from clothes');

    if(Auth::check()){

        return view('home', ['clothes' => $clothes]);

    } else {

        return view('guestHome', ['clothes' => $clothes]);

    }

}

public function logOut(){

    Session::flush();

    Auth::logout();

    return Redirect('home');

}

}

```

Kodingan dalam *controller* ini akan melakukan operasi yang berhubungan dengan autentikasi pengguna. Terdapat fungsi “login” agar pengguna dapat masuk ke beranda *web* dan “register” untuk melakukan pendaftaran sebelum masuk ke beranda. Fungsi “validateRegister” dan “validateLogin” akan memvalidasi data-data yang dimasukkan oleh *user*. Jika pendaftaran berhasil maka user akan diarahkan ke halaman *login* dan jika *login* berhasil maka pengguna akan diarahkan ke beranda. Untuk keluar dari beranda, fungsi “logout” digunakan agar pengguna dapat keluar dari *website* lalu diarahkan kembali ke halaman *login*.

CartController:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Session;

use App\Models\User;
use App\Models\Clothes;
use App\Models\Cart;
use App\Models\Checkout;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;

class CartController extends Controller{

    public function cart(Request $request){
```

```

$cart = DB::table('cart');

$cartClothes = $cart->join('clothes', 'clothes.id', '=',
'cart.clothes_id')->where('cart.user_id', '=', $request->user()->id)->get();

return view('cart', compact('cartClothes'));
}

public function addToCart(Request $request) {

    $clothes = DB::select('select * from clothes where id = ?',
[$request->input('clothes_id')]);

    $cart = Cart::where('clothes_id', $request->input('clothes_id'))->where('user_id',
$request->input('user_id'))->first();

    if($cart) {

        $cart->quantity = $cart->quantity + 1;

        $cart->update();

    } else {

        DB::insert('insert into cart (clothes_id, user_id, quantity) values (?, ?, 1)',
[$request->input('clothes_id'), $request->input('user_id')]);

    }

    return redirect()->action([ClothesController::class, 'clothesDetail'], ['name' =>
$clothes[0]->name])->with('success', 'Your clothes has been added to the cart.');
```

```

}

public function checkout(){

    return view('checkout');

}

```

```
public function doCheckout(Request $request){

    $request->validate([

        'name' => 'required|min:5',

        'phone' => 'required|digits:12',

        'address' => 'required',

    ]);

    $check = new Checkout;

    $check->name = $request->input('name');

    $check->phone = $request->input('phone');

    $check->address = $request->input('address');

    $check->save();

    return redirect()->route('home')->with('success', 'Checkout success');

}

public function destroy($cart_id){

    $cart = Cart::find($cart_id);

    $cart->delete();

    return redirect()->route('cart');

}

}
```

Kodingan *controller* ini akan mengatur berbagai operasi-operasi yang akan dilakukan oleh *user* pada saat dan pada saat melakukan checkout. *cart* digunakan untuk melihat halaman *cart*. Dimana *controllernya* akan meminta *user* id dan

clothes id yang dimasukkan ke cart. Lalu, ada function addToCart merupakan controller untuk menambahkan pakaian ke cart. Apabila pakaian dengan id yang sama sudah ada di cart, maka jumlah dari pakaian tersebut ditambahkan dengan 1. Jika data pakaiannya belum ada maka web akan memasukkannya ke cart.

Ada function checkout untuk melihat page checkout dan function untuk memvalidasi input yang dimasukkan oleh *user* pada saat melakukan checkout dan menyimpannya ke tabel checkout di database. Function destroy untuk menghapus pakaian yang ada di cart dengan menemukannya berdasarkan cart_idnya.

ClothesController.php:

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use App\Models\Clothes;

class ClothesController extends Controller {

    public function clothesDetail(Request $request) {

        $clothes = DB::select('select * from clothes where name = ?', [$request->name]);

        return view('clothes_detail', ['clothes' => $clothes, 'name' => $request->name, 'user' =>
$request->user()->id]);
    }
}
```

```

}

public function addClothes() {
    return view('add_clothes');
}

public function storeClothes(Request $request) {
    $request->validate([
        'name' => 'required|min:5',
        'type'  => 'required|min:1',
        'brand' => 'required|min:3',
        'color' => 'required|min:3',
        'size'  => 'required|min:1',
        'price' => 'required|numeric|min:1',
        'description' => 'required|max:255',
        'image' => 'required|image|mimes:jpg,png'
    ]);

    $clothes = new Clothes;

    $image = $request->file('image');
    $imageName = time().'.'.$image->getClientOriginalExtension();
    $image->storeAs('public/images', $imageName);
    $clothes->image = $imageName;
}

```



```
$clothes->name = $request->input('name');

$clothes->type = $request->input('type');

$clothes->brand = $request->input('brand');

$clothes->color = $request->input('color');

$clothes->size = $request->input('size');

$clothes->price = $request->input('price');

$clothes->description = $request->input('description');

$clothes->save();

return redirect('add_clothes')->with('message', 'Clothes successfully added');
}

public function search(Request $request){

    $clothes = Clothes::where("name", "LIKE", "%$request->search%")->simplePaginate();

    return view('search_clothes')->with('clothes', $clothes);
}
}
```

Terdapat beberapa fungsi di dalam ClothesController yang mengatur cara berkejanya model *Clothes* dalam aplikasi web.

- **clothesDetail**

Fungsi ini bertugas untuk mengambil detail dari pakaian tertentu berdasarkan nama. Ia melakukan query ke database menggunakan facade DB untuk memilih semua data dari tabel clothes yang cocok dengan nama yang diberikan.

- **addClothes**

Fungsi ini menyediakan tampilan formulir untuk menambahkan pakaian baru. Ketika dipanggil, fungsi ini hanya mengembalikan tampilan `add_clothes`, yang biasanya akan berisi formulir untuk input data pakaian.

- **storeClothes**

Fungsi ini menangani logika penyimpanan pakaian baru ke dalam database. Ia memulai dengan memvalidasi data permintaan yang diterima, memastikan bahwa semua field wajib ada dan sesuai dengan kriteria yang ditentukan (seperti panjang string dan jenis data). Data yang divalidasi kemudian digunakan untuk membuat instance baru dari model `Clothes`, diisi dengan data tersebut, dan disimpan ke dalam database.

- **search**

Fungsi ini memungkinkan pengguna untuk mencari pakaian berdasarkan kata kunci. Ia menggunakan Eloquent untuk melakukan query where dengan operator `LIKE` pada kolom `name` di tabel `clothes`.

WebController:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Hash;
use Session;
use App\Models\User;
use App\Models\Clothes;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
```

```
use Illuminate\Support\Facades\Storage;

class WebController extends Controller{

    public function home(Request $request){

        if($request->type){

            $type = $request->type;

        } else {

            $type = 'T-shirt';

        }

        $clothes = DB::select('select * from clothes where
type = ?', [$type]);

        if(Auth::check()){

            return view('home', ['clothes' => $clothes, 'type'
=> $type]);

        } else {

            return view('guestHome', ['clothes' => $clothes,
'type' => $type]);

        }

    }

}
```

Terdapat sebuah fungsi dalam WebController yang mengatur cara bekerjanya web dalam mengarahkan pengguna ke halaman beranda. Fungsi “home” bertugas untuk mengarahkan pengguna yang dinyatakan sebagai tamu atau belum melakukan *login* dan pengguna yang sudah melakukan login dan dinyatakan sebagai anggota dari *website*.

View

navbar.blade:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <title>@yield('title')</title>

    <link
href="https://fonts.googleapis.com/css2?family=Nunito:wght@400
;600;700&display=swap" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font
/bootstrap-icons.css" rel="stylesheet">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bo
otstrap.min.css" rel="stylesheet"
integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+
PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/boot
strap.bundle.min.js"
integrity="sha384-HwvvtgBNo3bZJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVI
nixGAoxmnlMuBnhbgrkm" crossorigin="anonymous"></script>

</head>
```

```
<body>

  <nav class="navbar navbar-light navbar-expand-lg mb-5"
style="background-color: #e5DFD8;">

    <div class="container">

      <a class="navbar-brand"></a>

      <div class="collapse navbar-collapse"
id="navbarNav">

        <ul class="navbar-nav">

          @guest

            <li class="nav-item">

              <a class="nav-link" href="{{
route('home') }}"><b><span style="color:
black;">Home</span></b></a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{
route('login') }}"><b><span style="padding-left: 940px; color:
black;">Login</span></b></a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{
route('register') }}"><b><span style="color:
black;">Register</span></b></a>

            </li>

          @else

            <li class="nav-item">
```

```

        <a class="nav-link" href="{{
route('home') }}"><b><span style="color:
black;">Home</span></b></a>

    </li>

    <li class="nav-item">

        <a class="nav-link" href="{{
route('search_clothes') }}"><b><span style="color:
black;">Search Clothes</span></b></a>

    </li>

    <li class="nav-item">

        <a class="nav-link" href="{{
route('cart') }}"><b><span style="color:
black;">Cart</span></b></a>

    </li>

    <li class="nav-item">

        <a class="nav-link" href="{{
route('logout') }}" style = "padding-left: 600px; color:
black;"><b><span style="color: black;">Logout</span></b></a>

    </li>

    <li class="nav-item" style="margin-top:
9px;">

        <p><b><span style="color: black;
padding-left: 50px; margin-right: 10px;">Welcome,
{{Auth::user()->name}}</span></b></p>

    </li>

    <li class="nav-item">

    </li>

```

```

                                @endguest

                                </ul>

                                </div>

                                </div>

                                </nav>

                                @yield('content')

</body>

</html>

```

Kodingan dari “navbar.blade” bertugas untuk membuat *navigation bar* pada setiap halaman *web*. Terdapat dua pilihan untuk *navigation bar* yang digunakan, apabila penggunaannya masih dinyatakan sebagai tamu atau belum menjadi anggota dari web maka *navigation bar* yang dimunculkan adalah berisi tombol *home* untuk kembali ke halaman beranda, *login* untuk masuk ke website sebagai anggotanya, *register* untuk melakukan pendaftaran.

Jika pengguna sudah merupakan bagian dari anggota web maka *navigation bar* yang ditampilkan adalah berisi tombol *home*, *search clothes*, *cart*, *logout*, dan kalimat “Welcome, (nama username)”. Fungsi-fungsi dari masing-masing tombol sebagai berikut:

- *home*: mengarahkan pengguna ke halaman beranda.
- *search clothes*: melakukan pencarian pakaian-pakaian.
- *cart*: memeriksa keranjang pengguna yang berisi pakaian-pakaian yang akan dibeli.
- *logout*: untuk keluar dari website dan dinyatakan sebagai tamu lagi.

register.blade

```
@extends('navbar')
```



```
<link rel="stylesheet" href="{{
asset('css/login_regist_style.css') }}">

@section('content')

<div class="card-body">

    <div class="formContainer col-md-3 mt-5">

        <form action="{{ route('register.validate') }}"
method="post">

            <div class="header">Register</div>

            @csrf

            <div class="form-group mt-3">

                <label for="email" class="form-label">Email
Address</label>

                <input type="text" name="email"
class="form-control" placeholder="Has to end with
'@gmail.com'" required>

                @if($errors->has('email'))

                    <span class="text-danger">{{
$errors->first('email') }}</span>

                @endif

            </div>

            <div class="form-group mt-3">

                <label for="name"
class="form-label">Username</label>

                <input type="text" name="name"
class="form-control" placeholder="Minimum 5 characters,
maximum 20 characters" required>
```

```

        @if($errors->has('name'))

            <span class="text-danger">{{
$errors->first('name') }}</span>

        @endif

    </div>

    <div class="form-group mt-3">

        <label for="password"
class="form-label">Password</label>

        <input type="password" name="password"
class="form-control" placeholder="Minimum 5 characters,
maximum 100 characters" required>

        @if($errors->has('password'))

            <span class="text-danger">{{
$errors->first('password') }}</span>

        @endif

    </div>

    <div class="form-group mt-3">

        <label for="confirm_password"
class="form-label">Confirm Password</label>

        <input type="password" name="confirm_password"
class="form-control" placeholder="Has to be same as previous
password" required>

        @if($errors->has('confirm_password'))

            <span class="text-danger">{{
$errors->first('confirm_password') }}</span>

        @endif

    </div>

    <div class="d-grid mx-auto mt-3">

```

```

        <button type="submit" class="btn btn-dark
btn-block">Register</button>

    </div>

    <p class="link mt-2">Already have an account? <a
href="{{ route('login') }}">Log in</a></p>

    </form>

</div>

</div>

@endsection('content')

```

View “register.blade” berfungsi menampilkan halaman pendaftaran yang akan digunakan oleh pengguna untuk melakukan pendaftaran. Dalam kodingannya, terdapat kodingan untuk memasukkan *email*, *username*, *password*, dan konfirmasi *password*. Apabila terdapat kesalahan maka *website* akan menampilkan pesan error yang menunjukkan perbaikan apa saja yang harus dilakukan oleh pengguna. Pada kodingan terdapat referensi *login_regist_style.css* untuk mengatur pengayaan pada halaman *register* ini.

login.blade:

```

@extends('navbar')

<link rel="stylesheet" href="{{
asset('css/login_regist_style.css') }}">

@section('content')

@if(session('success'))

```

```

    <div class="alert alert-success"> {{ session('success') }}
</div>

@elseif(session('fail_message'))

    <div class="alert alert-danger"> {{
session('fail_message') }} </div>

@endif

<div class="card-body">

    <div class="formContainer col-md-3 mt-5">

        <form action="{{ route('login.validate') }}"
method="post">

            <div class="header">Log in</div>

            @csrf

            <div class="form-group mt-3">

                <label for="email">Email address</label>

                <input type="text" name="email"
class="form-control" placeholder="Has to end with
'@gmail.com'" required>

                @if($errors->has('email'))

                    <span class="text-danger">{{
$errors->first('email') }}</span>

                @endif

            </div>

            <div class="form-group mt-3">

                <label for="password">Password</label>

                <input type="password" name="password"
class="form-control" placeholder="Minimum 5 characters,
maximum 100 characters" required>

```

```

        @if($errors->has('password'))

            <span class="text-danger">{{
$errors->first('password') }}</span>

        @endif

    </div>

    <div class="form-group mt-3">

        <div class="form-check">

            <input class="form-check-input"
type="checkbox" name="remember" id="remember">

            <label class="form-check-label"
for="remember"> Remember me </label>

        </div>

    </div>

    <div class="d-grid mx-auto mt-3">

        <button type="submit" class="btn btn-dark
btn-block">Login</button>

    </div>

    <p class="link mt-2">Don't have an account? <a
href="{{ route('register') }}">Sign up</a></p>

    </form>

</div>

</div>

@endsection('content')

```

Kodingan ini akan membuat tampilan halaman *log in* dimana pengguna dapat melakukan *log in* untuk masuk ke halaman beranda sebagai bagian anggota *website*. Pada kodingan dimasukkan referensi *login_regist_style.css* untuk mengatur penggayaan pada halaman *login* ini. Lalu, kodingan akan meminta pengguna untuk memasukkan alamat *email* dan *password* agar dan menekan tombol “login” agar dapat masuk ke halaman beranda. Jika terdapat kesalahan memasukkan input maka akan ditunjukkan pesan perbaikan.

People
Innovation
Excellence

home.blade:

```
@extends('navbar')

<link rel="stylesheet" href="{{ asset('css/home_style.css') }}">

@section('content')

@if(session('success'))

    <div class="alert alert-success"> {{ session('success') }}
</div>

@endif

<div class = "image-container">

</div>

<div class="title-container">

    <h1>Our Products</h1>

</div>
```

```
<div class="category-container">

    <h2>Choose by category</h2>

    <button class="btn btn-dark btn-block"> <a class =
"text-decoration-none text-white" href = "{{ route('home',
['type' => 'T-shirt']) }}"> T-shirt </a></button>

    <button class="btn btn-dark btn-block"> <a class =
"text-decoration-none text-white" href = "{{ route('home',
['type' => 'Hoodie']) }}"> Hoodie </a> </button>

    <button class="btn btn-dark btn-block"> <a class =
"text-decoration-none text-white" href = "{{ route('home',
['type' => 'Hat']) }}"> Hat </a> </button>

    <button class="btn btn-dark btn-block"> <a class =
"text-decoration-none text-white" href = "{{ route('home',
['type' => 'Pants']) }}"> Pants </a> </button>

    <br> <br>
</div>

<div class="box-container">

    @foreach($clothes as $clothing)

        <a class = "text-decoration-none" href="{{
route('clothes_detail', ['name' => $clothing->name]) }}">

            <div class="box">

                <div class="image-box">

                    <img src = "{{
asset('storage/images/' . $clothing->image) }}">

                </div>

                <p><b>{{ $clothing->name }}</b></p>

            </div>

        </a>

    </foreach>

</div>
```

```

        </div>

    </a>

@endforeach
</div>

@endsection('content')
```

Ini adalah kodingan untuk menunjukkan halaman beranda untuk pengguna yang sudah menjadi anggota *website*. Kodingan ini akan menampilkan tombol-tombol untuk memilih pakaian berdasarkan tipenya lalu menampilkan semua produk-produk pakaian berupa nama dan gambarnya. Lalu ditambahkan *navigation bar* pada bagian atasnya.

clothes_detail.blade:

```

@extends('navbar')

<link rel="stylesheet" href="{{ asset('css/detail_style.css') }}">

@section('content')

@if(session('success'))

    <div class="alert alert-success"> {{ session('success') }}
</div>

@endif
```



```
<div class="title-container">

    <h1 style="color: black">Clothes Detail</h1>

</div>

@guest

<div class="clothes-detail-container">

    @foreach($clothes as $clothing)

        <div class="image-container">

            <img src = "{{
asset('storage/images/' . $clothing->image) }}">

        </div>

        <div class="text-container">

            <h2 style="color: yellow;">{{ $name }}</h2>

            <br>

            <h2 style="color: yellow;">Type:</h2>

            <p>{{ $clothing->type }}</p>

            <h2 style="color: yellow;">Brand:</h2>

            <p>{{ $clothing->brand }}</p>

            <h2 style="color: yellow;">Color:</h2>

            <p>{{ $clothing->color }}</p>

            <h2 style="color: yellow;">Price:</h2>

            <p>Rp{{ $clothing->price }}.000</p>

            <h2 style="color: yellow;">Size:</h2>

            <p>{{ $clothing->size }}</p>

            <h2 style="color: yellow;">Description:</h2>
```

```

        <p>{{ $clothing->description }}</p>

    </div>

@endforeach

</div>

@else

<div class="clothes-detail-container">

    @foreach($clothes as $clothing)

        <div class="image-container">

            <img src = "{{
asset('storage/images/' . $clothing->image) }}">

        </div>

        <div class="text-container">

            <h2 style="color: yellow;">{{ $name }}</h2>

            <br>

            <h2 style="color: yellow;">Type:</h2>

            <p>{{ $clothing->type }}</p>

            <h2 style="color: yellow;">Brand:</h2>

            <p>{{ $clothing->brand }}</p>

            <h2 style="color: yellow;">Color:</h2>

            <p>{{ $clothing->color }}</p>

            <h2 style="color: yellow;">Price:</h2>

            <p>Rp {{ $clothing->price }}.000</p>

            <h2 style="color: yellow;">Size:</h2>

            <p>{{ $clothing->size }}</p>

```

```

        <h2 style="color: yellow;">Description:</h2>

        <p>{{ $clothing->description }}</p>

        <form action="{{ route('cart.add') }}"
method="post" enctype="multipart/form-data">

            @csrf

            <input type="hidden" name="clothes_id"
value="{{ $clothing->id }}">

            <input type="hidden" name="user_id" value="{{
$user }}">

            <input type="submit" class="button" value="Add
to Cart">

        </form>

    </div>

@endforeach
</div>

@endguest

@endsection('content')

```

Kodingan ini akan menampilkan informasi mengenai pakaian seperti menunjukkan nama, tipe, warna, deskripsi, ukuran yang tersedia, dan merek. Terdapat dua versi, yang pertama untuk pengguna tidak terdapat tombol “addToCart” untuk memasukkan pakaian ke keranjang. Tetapi selain tamu yaitu anggota dari *website* terdapat tombol “addToCart”.

search_clothes.blade:

```
@extends('navbar')
```

```
<link rel="stylesheet" href="{{ asset('css/search_style.css')
}}">

@section('content')

<h1 style="color: black; margin-left: 50px; text-align:
center; margin-bottom: 40px;">Search Clothes</h1>

<div class="search-container">

    <form class="d-flex" action="{{ route('search_clothes')
}}">

        <input class="form-control me-2" type="search"
placeholder="Enter clothes name" aria-label="Search"
name="search">

        <button class="search_button btn-outline-success"
type="submit">Search</button>

    </form>
</div>

@if($clothes->isEmpty())

    <div class="empty-container">

        <p style="color: white;">The clothes is not
available.</p>

    </div>

@else

<div class="content">
```

```
@foreach($clothes as $clothing)

    <a class = "text-decoration-none" href="{{
route('clothes_detail', ['name' => $clothing->name]) }}">

        <div class="clothes-container">

            <div class="image-container">

                <img src = "{{
asset('storage/images/' . $clothing->image) }}">

            </div>

            <div class="text-container">

                <h2 style="color: yellow;">{{
$clothing->name }}</h2>

                <br>

                <h2 style="color: yellow;">Type:</h2>

                <p>{{ $clothing->type }}</p>

                <h2 style="color: yellow;">Brand:</h2>

                <p>{{ $clothing->brand }}</p>

                <h2 style="color: yellow;">Price:</h2>

                <p>Rp {{ $clothing->price }}.000</p>

                <h2 style="color:
yellow;">Description:</h2>

                <p>{{ $clothing->description }}</p>

            </div>

        </div>

    </a>

@endforeach

</div>
```

```
@endif

@endsection('content')
```

People
Innovation
Excellence

Kodingan bertugas untuk menampilkan halaman pencarian makanan berdasarkan nama yang dimasukkan oleh pengguna. Apabila nama yang diketik oleh pengguna tidak ada, maka akan ditampilkan kotak bertulis tidak ada pakaian yang dicari. Jika ada maka *website* akan menampilkan makanan yang sesuai dengan nama tersebut.

cart.blade:

```
@extends('navbar')

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGChEKrQN+
PtmoHDEXuppvndJzQIu9" crossorigin="anonymous">

<link rel="stylesheet" href="{{ asset('css/cart_style.css')
}}">

@section('content')

<div class="header">

    <h1>Your Cart</h1>

</div>

@if($cartClothes->isEmpty())
```

```

<table>

    <thead>

        <tr>

            <th style="color: yellow">Clothes Name</th>

            <th style="color: yellow">Price</th>

            <th style="color: yellow">Quantity</th>

            <th style="color: yellow">Total</th>

        </tr>

    </thead>

    <tbody>

    </tbody>

</table>

@else

    <table>

        <thead>

            <tr>

                <th style="color: yellow">Clothes Name</th>

                <th style="color: yellow">Price</th>

                <th style="color: yellow">Quantity</th>

                <th style="color: yellow">Total</th>

                <th> </th>

            </tr>

        </thead>

        <tbody>

            @php($sum = 0)

```

```

        @foreach($cartClothes as $cart)

        <tr>

            <td>{{ $cart->name }}</td>

            <td>Rp {{ $cart->price }}.000</td>

            <td>{{ $cart->quantity }}</td>

            <td>Rp {{ $cart->price * $cart->quantity
        }}.000</td>

            <td>

                <form action="{{ route('carts.destroy',
$cart->cart_id) }}" method="post">

                    @csrf

                    @method('DELETE')

                    <button type="submit"
class="remove-button">Delete</button>

                </form>

            </td>

        </tr>

        @php($sum += ($cart->price * $cart->quantity))

    @endforeach

</tbody>

</table>

<div class="container">

    <h2 style="color: black; margin-bottom: 20px">Your
total price: Rp {{ $sum }}.000</h2>

    <div class="button-container">

```



```

        <button type="submit" class="button"
onclick="window.location='{{ route('checkout')
}}'">Checkout</button>

    </div>

</div>

@endif

@endsection('content')

```

Kodingan ini bertugas untuk menampilkan halaman keranjang berisi daftar-daftar nama pakaian yang sudah dimasukkan ke keranjang beserta harga, jumlah, dan total harga. Apabila belum ada pakaian yang dimasukkan maka *website* menunjukkan pesan bahwa keranjang masih kosong. Lalu terdapat daerah untuk menampilkan total keseluruhan harga dan tombol “checkout” untuk mengarahkan pengguna ke halaman *checkout*.

checkout.blade:

```

@extends('navbar')

<link rel="stylesheet" href="{{
asset('css/checkout_style.css') }}">

@section('content')

<div class="header">

    <h1 style="color: black">Checkout</h1>

</div>

```

```
<div class="container">

  <div class="row justify-content-center">

    <div class="col-md-6">

      <form action="{{ route('cart.checkout') }}"
method="post" enctype="multipart/form-data">

        @csrf

        <div class="form">

          <div class="form-group mt-3">

            <label for="name">Name: </label>

            <input type="text" name="name"
placeholder="Enter your name, minimum 5 characters"
class="form-control">

          </div>

          <div class="form-group mt-3">

            <label for="phone">Phone number:
</label>

            <input type="phone" name="phone"
placeholder="Enter your phone number, has to be 12 digits"
class="form-control">

          </div>

          <div class="form-group mt-3">

            <label for="address">Home address:
</label>

            <input type="text" name="address"
placeholder="Enter your home address" class="form-control">

          </div>

        </div>

        <div class="button">
```

```

        <button type="submit"
class="checkout-button">Checkout</button>

    </div>

</form>

</div>

</div>

</div>

@endsection('content')

```

Kodingan yang bertugas untuk menampilkan halaman *checkout* untuk melakukan pembayaran terhadap pakaian-pakaian yang dimasukkan ke keranjang. Website meminta pengguna untuk memasukkan nama, nomor telepon, dan alamat rumah. Setelah semua diisi terdapat tombol “submit” untuk melakukan *checkout*.

add_clothes.blade:

```

<!DOCTYPE HTML>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/css/bo
otstrap.min.css" rel="stylesheet">

```

```
<link rel="stylesheet" href="{{ asset('css/add_style.css')
}}">

</head>

<body>

    @if ($errors->any())

        <div>

            <ul>

                @foreach ($errors->all() as $error)

                    <li>{{ $error }}</li>

                @endforeach

            </ul>

        </div>

    @endif

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md-6">

                <div class="header">

                    <h1>Add Clothes</h1>

                </div>

                <form action="{{ route('store_clothes') }}"
method="post" enctype="multipart/form-data">

                    @csrf

                    <div class="form">

                        <div class="form-group mt-3">
```

```

<label for="name">Clothes
Name</label>

<input type="text" name="name"
placeholder="Enter the clothes name" class="form-control">

</div>

<div class="form-group mt-3">

<label
for="brief_description">Clothes Type</label>

<input type="text" name="type"
placeholder="Enter the clothes type" class="form-control">

</div>

<div class="form-group mt-3">

<label for="about">Clothes
Brand</label>

<input type="text" name="brand"
placeholder="Enter the clothes brand" class="form-control">

</div>

<div class="form-group mt-3">

<label for="type">Clothes
Color</label>

<input type="text" name="color"
placeholder="Enter the clothes color" class="form-control">

</div>

<div class="form-group mt-3">

<label for="price">Clothes
Size</label>

<input type="text" name="size"
placeholder="Enter the clothes size" class="form-control">

```

```

        </div>

        <div class="form-group mt-3">

            <label for="price">Clothes
Price</label>

            <input type="text" name="price"
placeholder="Enter the clothes price" class="form-control">

        </div>

        <div class="form-group mt-3">

            <label for="image">Clothes
Description</label>

            <input type="text"
name="description" placeholder="Enter the clothes description"
class="form-control">

        </div>

        <div class="form-group mt-3">

            <label for="image">Clothes
Image</label>

            <input type="file" name="image"
class="form-control">

        </div>

    </div>

    <div class="button">

        <button type="submit">Submit</button>

    </div>

</form>

</div>

</div>

```

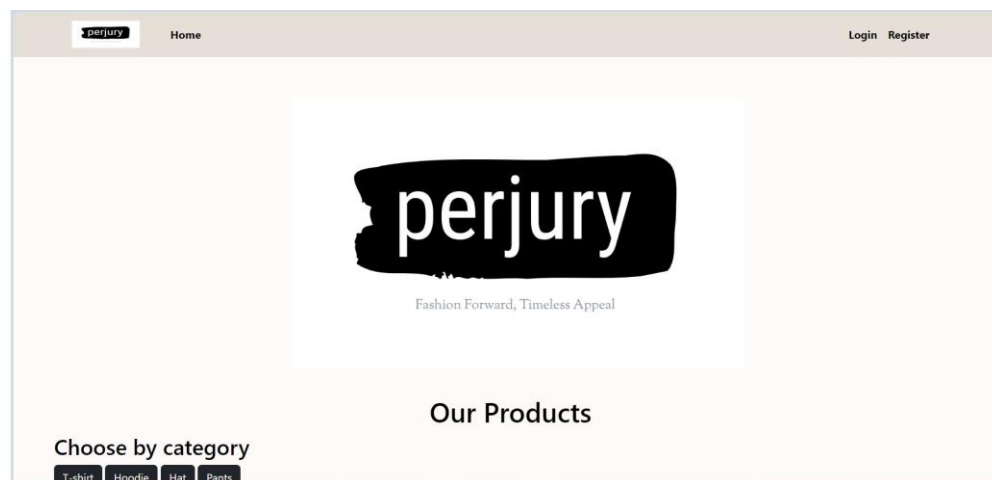
```
</div>  
  
</body>  
  
</html>
```

People
Innovation
Excellence

Kodingan ini hanya dapat digunakan oleh yang punya website untuk memasukkan data-data pakaian yang akan dijual.

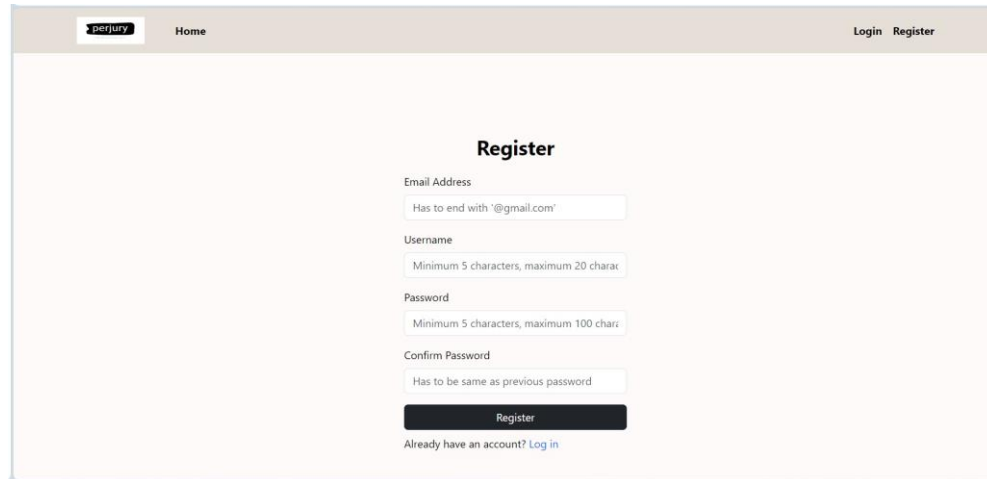
UI website

Home page (tamu / pengguna belum masuk atau mendaftar):



Register page:

Halaman utama ini disajikan untuk pengunjung yang belum mendaftarkan diri ke dalam sistem database. Perbedaan utamanya ada dalam navigation bar yang diatas dimana tidak ada *greeting* untuk *user* namun ada tombol login dan register yang menggantikannya. Halaman ini bertujuan untuk memberikan kesan pertama yang informatif dan menarik bagi pengunjung, serta mengarahkan mereka untuk melakukan pendaftaran



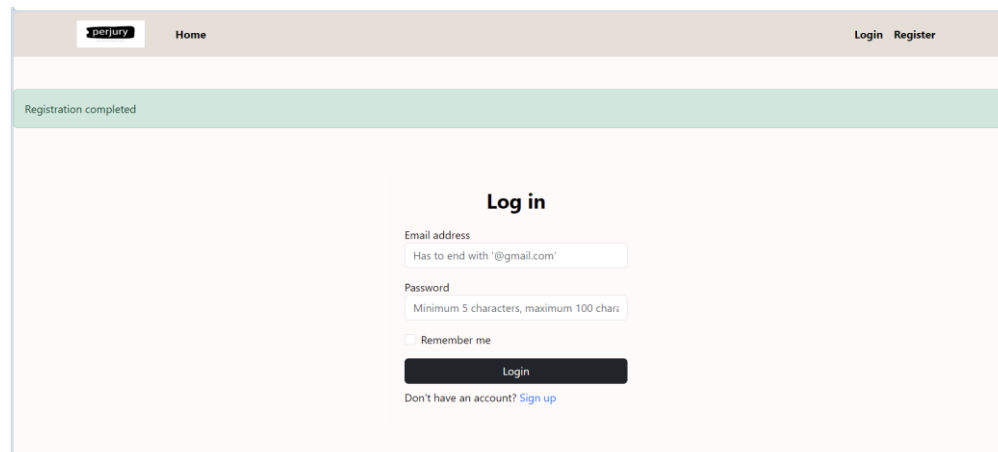
Pada halaman ini *user* dapat melakukan registrasi akun baru yang nantinya akan digunakan pada halaman login untuk masuk ke web Perjury. pada halaman Register ini *user* diminta untuk memasukan beberapa data diri seperti email address, username, password, dan konfirmasi password.

email berisikan alamat email dari masing-masing *user*, username berisikan nama yang *user* gunakan untuk nantinya login kedalam web Perjury ini, password merupakan kata sandi yang nantinya digunakan untuk login, dan terakhir adalah confirm password untuk memastikan password yang diisi sama dengan password sebelumnya.

terdapat juga beberapa syarat yang perlu diperhatikan pada saat *user* melakukan pengisian data diri seperti pada bagian email *user* memerlukan data email yang mengandung kata “@gmail.com”, pada bagian username *user* perlu memasukan nama dengan minimal 5 karakter dan maksimal 20 karakter, pada bagian “password” *user* perlu memasukan minimal 5 karakter dan maksimal 100 karakter, dan terakhir pada bagian konfirmasi password *user* perlu memasukan kata - kata yang sama dengan password sebelumnya.

jika *user* sudah mempunyai akun, *user* dapat langsung menekan “Log in” yang ada pada paling bawah halaman untuk langsung menuju ke halaman login. jika belum punya akun maka *user* dapat memasukan data diri dan menekan tombol “Register” yang ada untuk membuat akun.

Log in page:



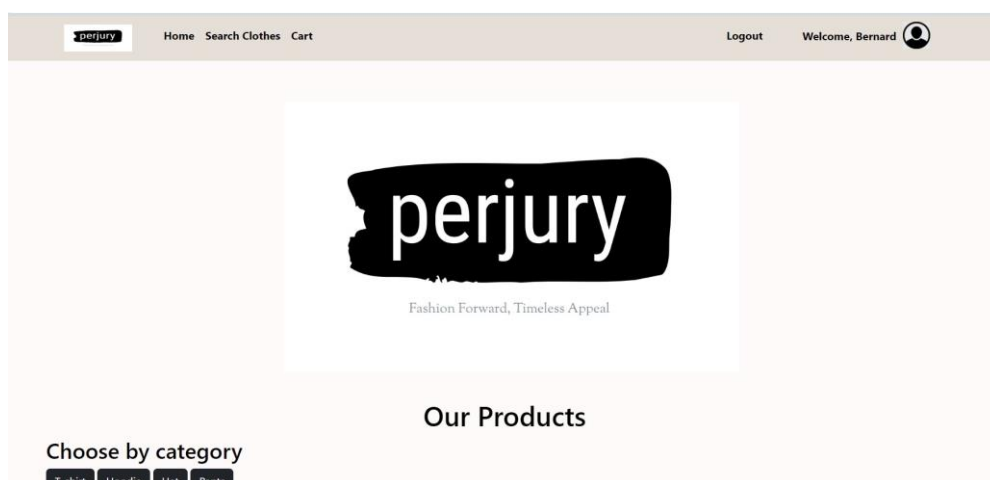
The screenshot shows the login page of the Perjury website. At the top, there is a navigation bar with the Perjury logo, a 'Home' link, and 'Login' and 'Register' links. Below the navigation bar, a green banner indicates 'Registration completed'. The main content area features a 'Log in' section with the following elements:

- Email address:** A text input field with a placeholder 'Has to end with '@gmail.com''.
- Password:** A text input field with a placeholder 'Minimum 5 characters, maximum 100 charz'.
- Remember me:** A checkbox labeled 'Remember me'.
- Login:** A dark button labeled 'Login'.
- Don't have an account?** A link labeled 'Sign up' in blue text.

Pada halaman ini *user* yang telah memiliki akun dapat melakukan login untuk masuk ke halaman utama Perjury. *User* yang sudah memiliki akun dapat langsung memasukan email dan juga password yang sebelumnya sudah dibuat pada halam Register dan menekan tombol “Login” untuk masuk ke halaman utama Perjury. *User* juga bisa mencentang *checkbox* yang bertuliskan Remember me agar setiap kali *user* membuka website Perjury *user* tidak perlu melakukan login kembali karena data diri sudah tersimpan.

Jika *user* belum memiliki akun maka *user* dapat menekan tulisan berwarna biru di paling bawah halaman yang bertuliskan “Sign up” untuk membuat akun.

Home page:

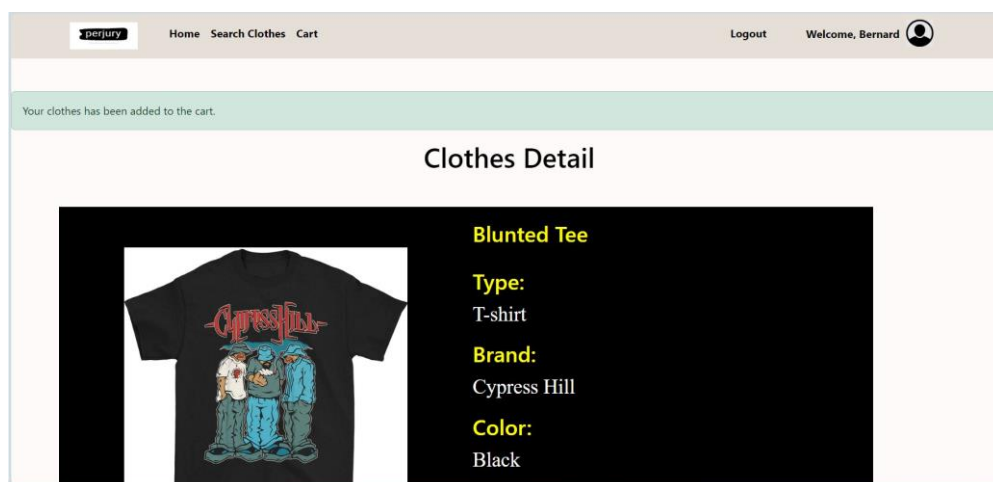


Pada halaman Home Page ini *user* dapat melakukan berbagai macam hal. *User* dapat melakukan search produk yang tersedia dengan menekan tombol “search” yang ada di halaman NavBar. *User* juga dapat mencari kategori produk yang tersedia dengan menekan berbagai macam kategori yang tersedia di bagian kiri bawah halaman seperti T-shirt, Hoodie, Hat, dan Pants. *user* juga dapat menuju ke halaman Cart untuk melihat produk apa saja yang ingin dibeli. *User* juga dapat melakukan Logout akun dengan cara menekan tombol “Logout” yang ada di bagian kanan atas halaman.



Jika di scroll kebawah *user* dapat melihat beberapa produk yang tersedia pada website Perjury ini. halaman ini menampilkan semua produk kita seperti contoh di atas yaitu pada kategori Hoodie. Tidak hanya hoodie, ada berbagai macam juga produk yang dapat dilihat.

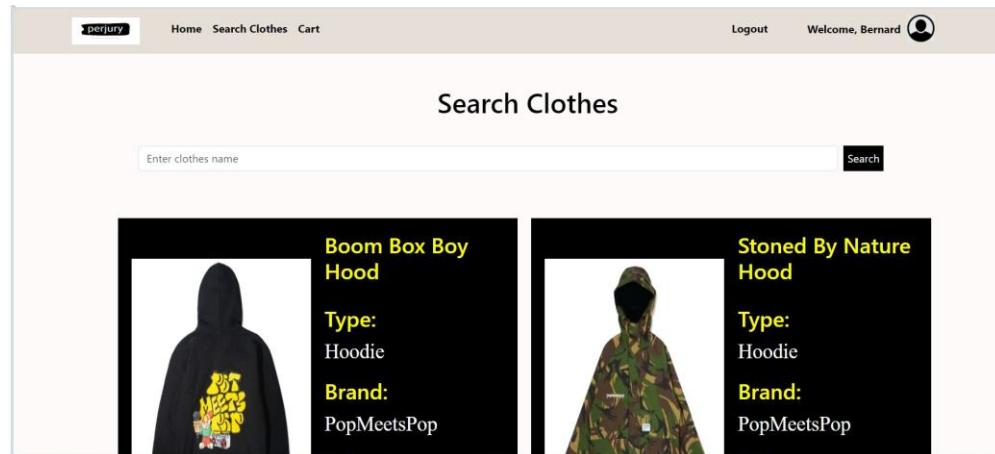
Clothes Detail page:



Jika *user* menekan salah satu produk yang tersedia pada website Perjury, *user* akan diteruskan untuk masuk ke halaman Clothes Detail. Pada halaman ini *User* dapat melihat berbagai macam informasi dari produk yang ditekan sebelumnya. *User* dapat melihat tipe produk, brand produk, warna produk, harga produk, ukuran produk yang tersedia, dan deskripsi dari produk itu sendiri.

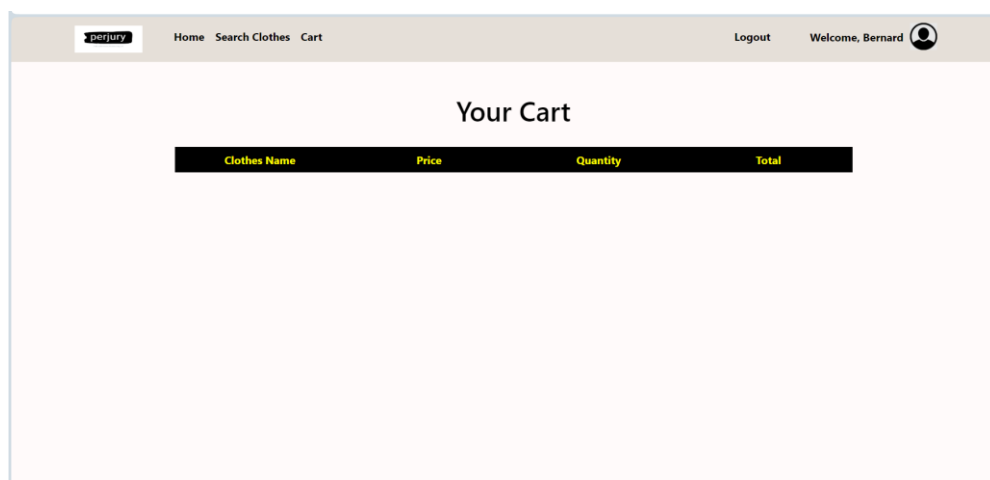
Pada halaman Clothes Detail ini *user* tidak hanya dapat melihat informasi mengenai produk yang dipilih namun *user* juga dapat melakukan pemesanan dengan cara menekan tombol “add to cart” yang ada di bagian paling bawah dari informasi produk tersebut. Jika *user* menekan tombol tersebut maka produk akan otomatis masuk ke dalam “cart” *user* untuk nantinya dapat melanjutkan proses pemesanan.

Search Clothes Page:



Tampilan diatas merupakan tampilan pada halaman Search Clothes. *User* akan masuk ke halaman ini jika *user* menekan tombol “Search Clothes” yang ada di bagian NavBar. Pada halaman ini *user* dapat melakukan pencarian produk yang tersedia di Perjury. *User* hanya perlu memasukan kata kunci pada bagian search lalu menekan tombol search yang ada di bagian kanan, jika produk ada maka web akan langsung menampilkan produk produk yang ada dibawahnya.

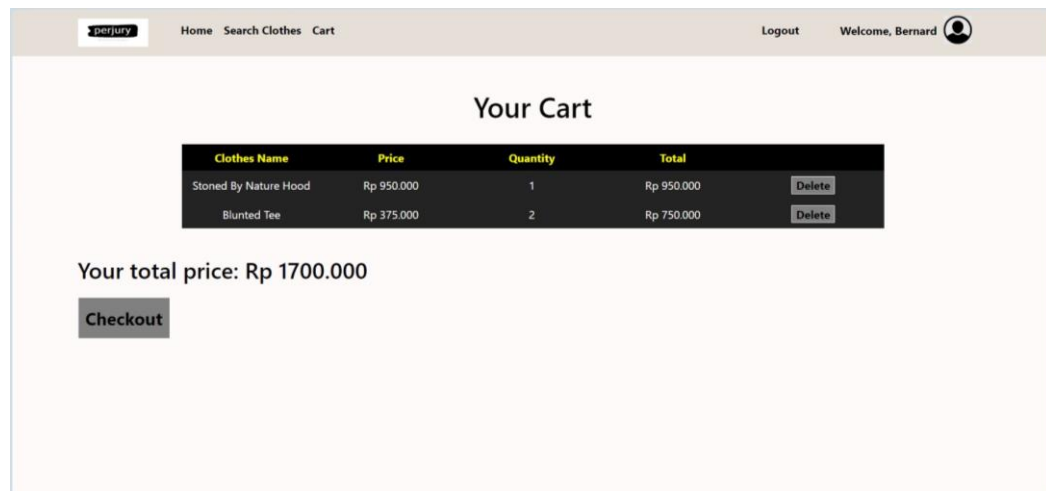
Cart page (kosong):



Jika *user* menekan tulisan “Cart” pada bagian NavBar maka *user* akan diteruskan ke halaman “Your Cart” seperti gambar diatas. Jika *user* belum menambahkan produk sama sekali kepada cart maka tampilan akan kosong seperti itu.

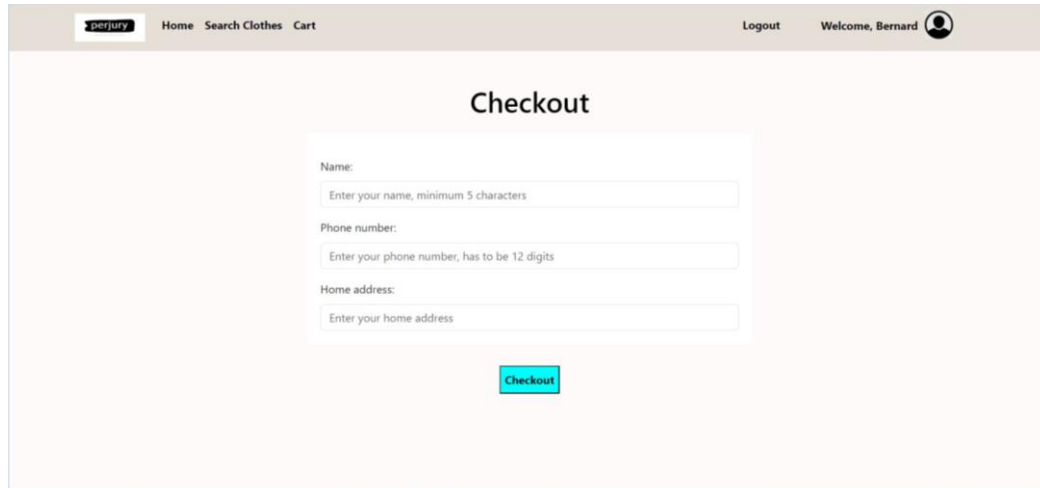
Cart page (sudah diisi):

People
Innovation
Excellence



Menekan tombol remove di tabel sebelahnya akan menghapus pakaian yang ada di tabel cart. Apabila *user* memasukkan pakaian yang sama ke keranjang, maka jumlahnya akan ditambahkan dengan 1.

Checkout Page:

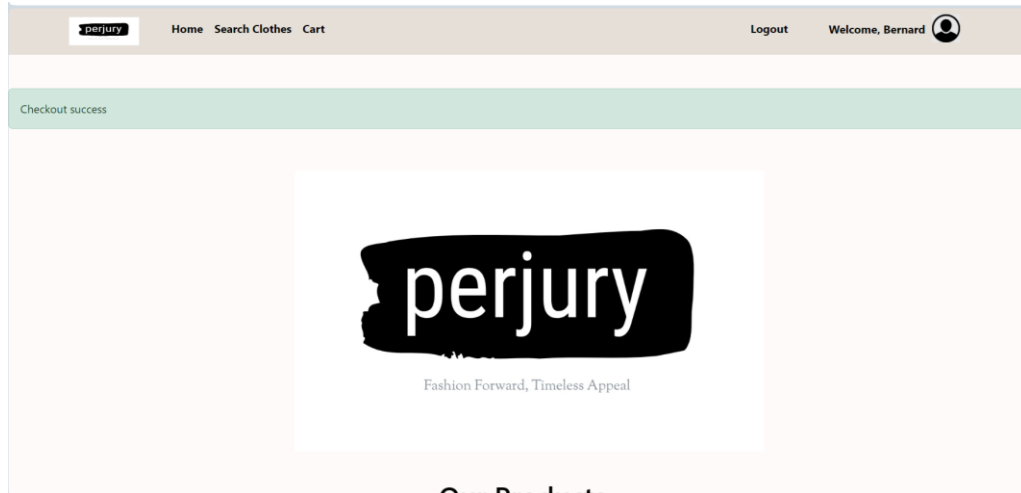


The screenshot shows a web application interface for a checkout process. At the top, there is a navigation bar with links: "Home", "Search Clothes", and "Cart". On the right side of the navigation bar, there are links for "Logout" and "Welcome, Bernard" next to a user profile icon. The main heading of the page is "Checkout". Below the heading, there is a form with three input fields: "Name:" with a placeholder "Enter your name, minimum 5 characters", "Phone number:" with a placeholder "Enter your phone number, has to be 12 digits", and "Home address:" with a placeholder "Enter your home address". At the bottom of the form, there is a blue button labeled "Checkout".

Setelah *user* selesai dan setuju dengan produk - produk yang ada pada halaman “Cart” sebelumnya maka *user* dapat menekan tombol “Checkout” dan akan diteruskan ke halaman Checkout seperti gambar diatas.

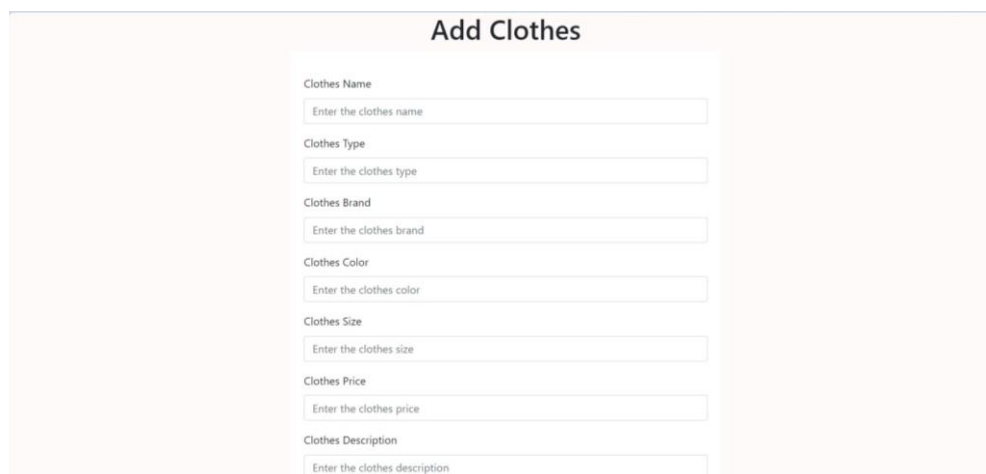
Pada halaman ini *User* dapat melanjutkan pesanan dengan cara mengisi beberapa formulir sebagai pendataan yaitu Nama yang harus lebih dari 5 karakter, Nomor HP yang harus berjumlah 12 digit dan harus angka, dan juga alamat rumah tinggal. setelah *user* sudah mengisi seluruh data yang diperlukan *user* dapat melakukan checkout dengan cara menekan tombol “Checkout” berwarna biru yang ada di paling bawah halaman.

Setelah checkout:



Setelah *User* selesai melakukan Checkout produk maka *user* akan langsung dialihkan kembali ke halaman utama namun dengan adanya pemberitahuan “Checkout Success” berwarna hijau yang akan muncul dibagian atas halaman seperti pada gambar yang ada diatas.

Add clothes (hanya bisa diakses oleh pemilik website):



The screenshot shows a form titled 'Add Clothes' for adding new products to the website. The form contains several input fields, each with a label and a placeholder text:

- Clothes Name:** Enter the clothes name
- Clothes Type:** Enter the clothes type
- Clothes Brand:** Enter the clothes brand
- Clothes Color:** Enter the clothes color
- Clothes Size:** Enter the clothes size
- Clothes Price:** Enter the clothes price
- Clothes Description:** Enter the clothes description

KESIMPULAN DAN SARAN

Dalam pengembangan situs ini, kami berfokus pada beberapa fungsi kunci seperti sistem login dan registrasi, pengelolaan produk, pencarian, penambahan item ke keranjang belanja, dan proses checkout. Kami juga memastikan bahwa situs tersebut dapat diakses dengan baik di berbagai perangkat, memenuhi standar desain responsif yang merupakan salah satu kriteria penilaian utama dalam mata kuliah ini.

Kesimpulan dari proyek ini adalah bahwa pembuatan situs web merupakan proses yang kompleks dan dinamis. Kami sebagai mahasiswa telah berhasil menerapkan konsep-konsep pemrograman web yang meliputi front-end dan back-end development. Kami juga telah belajar untuk bekerja secara kolaboratif dalam tim, mengelola waktu dan sumber daya dengan efektif, serta mengatasi berbagai hambatan teknis yang muncul selama proses pengembangan.

DAFTAR PUSTAKA

<https://www.uc.ac.id/fdb/membangun-bisnis-fashion-online-yang-sukses-tips-dan-strategi/>

<https://umkmrakyat.id/kegunaan-website-umkm-bagi-bisnis-fashion/>

<https://www.parapuan.co/read/533510670/7-rekomendasi-situs-fashion-untuk-menemukan-berbagai-inspirasi-gaya>

<https://id.techinasia.com/gila-fashion-kunjungi-9-website-ecommerce-fashion-terbaik-di-indonesia>