

AFH CAFE

PRESENTASI PROYEK AKHIR WEBSITE



DISUSUN OLEH KELOMPOK 1



Anggota Kelompok

01

Arya Dwipa Mukti
(Dokumentasi)

02

Fandhi Syahru Rishaleh
(Desainer)

03

Haris Cahyana
(Programming)

Content

- 01 Pengertian
- 02 Halaman Login
- 03 Halaman Home
- 04 Halaman Menu
- 05 Halaman Pesanan
- 06 Cetak Nota

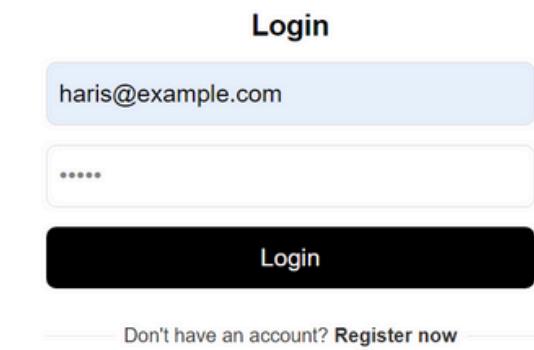


AFH CAFE

AFH CAFE adalah website yang memungkinkan kasir untuk menambahkan, menghapus dan mengedit pesanan/menu dalam halaman menu. kemudian mencetak nota dan mengakses pilihan menu yang tersedia pada halaman pesanan. Website ini memungkinkan untuk memudahkan kasir untuk mengelola pesanan.



Halaman Login



A screenshot of a login interface. At the top center, it says "Login". Below that are two input fields: the first contains the email "haris@example.com" and the second contains four dots representing a password. At the bottom is a black rectangular button with the word "Login" in white. Below the button, a small line of text reads "Don't have an account? [Register now](#)".

Formulir login terdiri dari dua field input: satu untuk alamat email dengan contoh 'haris@example.com', dan satu lagi untuk kata sandi, yang ditampilkan dalam bentuk titik-titik untuk menjaga keamanan. Di bawahnya, ada tombol hitam dengan teks 'Login' untuk mengirimkan informasi login.

```
<html lang="en">
<body>

<div class="login-container">
    @if(Session::has('error'))
        <div class="alert">
            {{ Session::get('error') }}
        </div>
    @endif

    <form action="{{route('actionlogin')}}" method="post">
        @csrf
        <div class="div">
            <div class="div-2">Login</div>
            <input class="div-3" name="email" type="text" placeholder="Email" required="required" />
            <input class="div-4" name="password" type="password" placeholder="Password" required="required" />
            <input class="div-5" type="submit" value="Login" />
        <div class="div-6">
            <div class="div-7"></div>
            <div class="div-8">
                Don't have an account?
                <span style="font-weight: bold; color: #007bff; cursor: pointer;">
                    <a href="register">Register</a>
                </span>
            </div>
        </div>
    </form>
</div>
```

PERANCANGAN HALAMAN LOGIN

01

Halaman login menampilkan form login yang memungkinkan pengguna untuk memasukkan email dan password mereka. Jika ada kesalahan login, pesan error akan ditampilkan di bagian atas form. Setelah pengguna memasukkan informasi mereka dan mengklik tombol login, data dikirim ke route actionlogin dengan metode POST untuk diproses. Token CSRF disertakan untuk keamanan form.

PERANCANGAN HALAMAN LOGIN

```
jin.blade.php x
resources > views > login.blade.php
<html lang="en">
<body>
<div class="login-container">
    @if(Session::has('error'))
        <div class="alert alert-danger" role="alert">
            {{ Session::get('error') }}
        </div>
    @endif
```

```
<form action="{{route('actionLogin')}}" method="post">
    @csrf
    <div class="div">
        <div class="div-2">Login</div>
        <input class="div-3" name="email" type="text" placeholder="Email" required="required" />
        <input class="div-4" name="password" type="password" placeholder="Password" required="required" />
        <input class="div-5" type="checkbox" checked="checked" /> Remember me
        <div class="div-6">
            <div class="div-7"></div>
            <div class="div-8">
                Don't have an account?
                <span style="font-weight: bold; color: #007bff; cursor: pointer;">
                    <a href="#">register</a>
                </span>
            </div>
        </div>
    </div>
```

02

03

04

login()

Fungsi login() akan mengarahkan pengguna ke halaman utama jika sudah masuk, jika belum, mereka akan diarahkan ke halaman login.

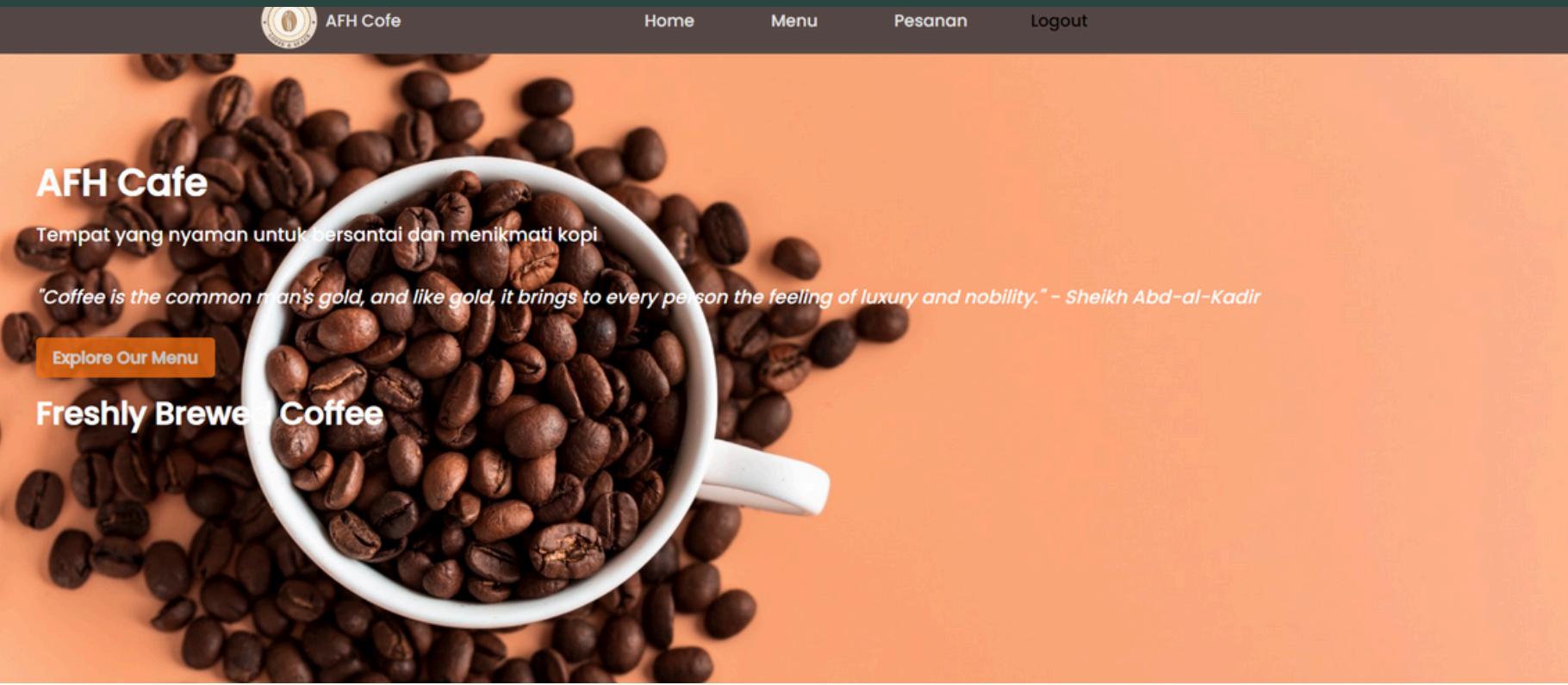
actionLogin()

Fungsi actionLogin() akan mengelola proses login pengguna dengan memeriksa email dan kata sandi yang dimasukkan. Jika informasi yang dimasukkan benar, pengguna akan diarahkan ke halaman utama, jika tidak, pesan kesalahan akan ditampilkan dan pengguna akan diarahkan kembali ke halaman login.

actionLogout()

Fungsi actionLogout() mengelola proses logout pengguna dengan menghapus sesi otentifikasi dan mengarahkan mereka kembali ke halaman login.

Halaman Home



Di bagian header, Terdapat logo AFH Cafe di sebelah kiri atas. Di sebelah kanan logo, terdapat menu navigasi dengan opsi 'Home,' 'Menu,'

'Pesanan,' dan 'Logout' untuk memudahkan pengguna menjelajahi situs.

Tepat di bawahnya, terdapat tombol aksi 'Explore Our Menu' yang berwarna oranye, kontras dengan latar belakang, menarik perhatian pengguna untuk mengeklik dan melihat menu kafe.

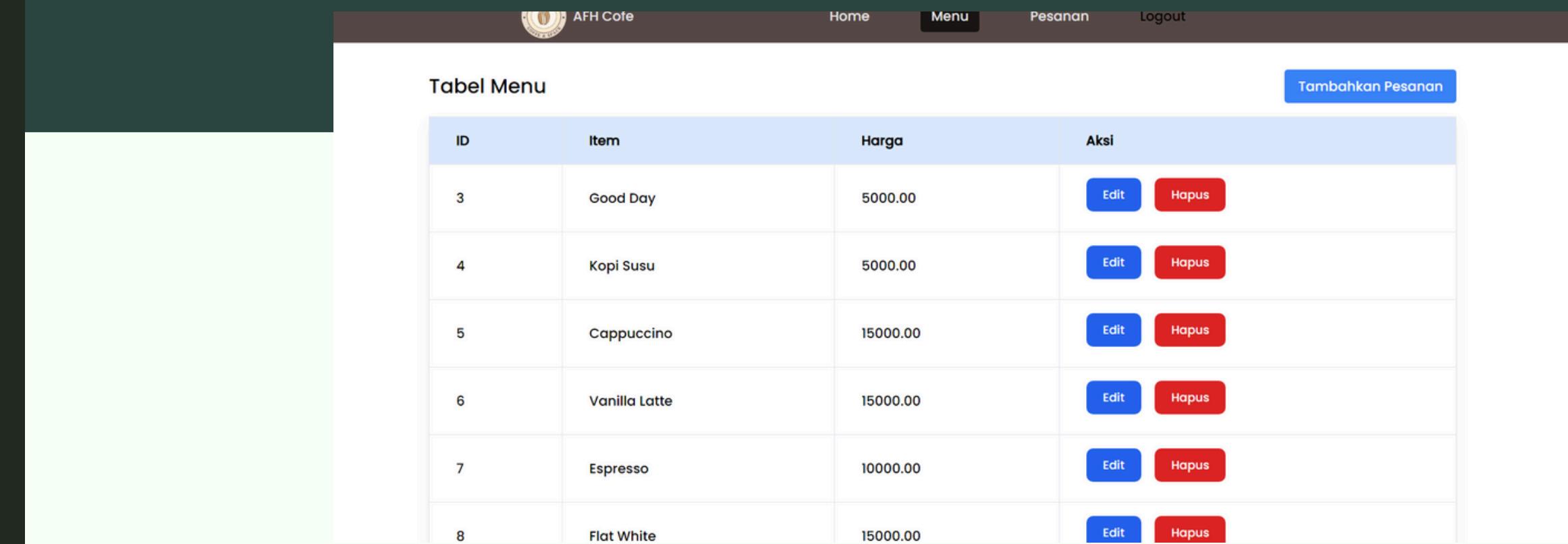
```
div class="div-4">
<div class="div-5">Selamat Datang di Sistem Informasi Pustaka Universitas Binaan Indonesia<br/>
<div class="div-6">
  Silahkan daftar terlebih dahulu!
</div>
<div class="div-7">
  <a href="login">
    <div class="div-8">Masuk</div>
  </a>
  <a href="register">
    <div class="div-9">Daftar</div>
  </a>
</div>
</div>
<div>
```

01

PERANCANGAN HALAMAN HOME

Pada halaman home, memanfaatkan tailwind untuk membuat desain teks dan tombol. Selain itu pada tombol Explore our menu menggunakan class animate dari tailwind untuk membuat efek animasi.

Halaman Menu



The screenshot shows a web application interface for managing a menu. At the top, there is a dark header bar with the logo "AFH Cofe" and navigation links: Home, Menu (which is highlighted in black), Pesanan, and Logout. Below the header is a table titled "Tabel Menu". The table has four columns: ID, Item, Harga, and Aksi. It contains six rows of data:

ID	Item	Harga	Aksi
3	Good Day	5000.00	Edit Hapus
4	Kopi Susu	5000.00	Edit Hapus
5	Cappuccino	15000.00	Edit Hapus
6	Vanilla Latte	15000.00	Edit Hapus
7	Espresso	10000.00	Edit Hapus
8	Flat White	15000.00	Edit Hapus

A blue button labeled "Tambahkan Pesanan" is located at the top right of the table area.

Pada Halaman menu, Pengguna dapat memesan menu dengan pilihan yang tersedia, terdapat empat kolom utama: ID, Item, Harga, dan Aksi. Kolom ID menampilkan nomor unik untuk setiap item menu, sementara Kolom Item mencantumkan nama-nama minuman yang tersedia di kafe ini. Harga dari setiap minuman ditampilkan dalam Kolom Harga dalam satuan rupiah. Terakhir, Kolom Aksi menyediakan dua tombol untuk setiap item: tombol "Edit" berwarna biru untuk mengubah item atau harga, dan tombol "Hapus" berwarna merah untuk menghapus item dari menu.

Edit Menu & Tambah Menu

Edit Menu

Item

Harga

Update **Cancel**

Create New Menu

Item

Harga

Create **Cancel**

```
7  
8 class MenuController extends Controller  
9 {  
10    /**  
11     * Display a listing of the resource.  
12     */  
13    public function index()  
14    {  
15        $menus = Menu::all()  
16        return view('menus/index', compact('menus'));  
17    }  
18  
19    /**  
20     * Show the form for creating a new resource.  
21     */  
22    public function create()  
23    {  
24        return view('menus/create');  
25    }  
26  
27    /**  
28     * Store a newly created resource in storage.  
29     */  
30    public function store(Request $request)  
31    {  
32        // Validation rules for the menu data  
33        $validatedData = $request->validate([  
34            'name' => 'required|string|max:255',  
35            'url' => 'required|url',  
36            'icon' => 'string|nullable',  
37            'parent_id' => 'integer|nullable',  
38        ]);  
39  
40        // Create the new menu item  
41        $menu = Menu::create($validatedData);  
42  
43        // Redirect back to the index page with success message  
44        return redirect()->back()->with('success', 'New menu item created successfully!');
```

PERANCANGAN HALAMAN MENU

index()

Menampilkan daftar semua menu yang ada. Ini mengambil semua data menu dari model Menu dan meneruskannya ke tampilan 'menus/index'.

create()

create(): Menampilkan formulir untuk membuat menu baru. Ini hanya mengembalikan tampilan 'menus/create'.

store(Request \$request)()

Menyimpan menu baru yang dibuat ke dalam penyimpanan. Pertama, metode ini memvalidasi input yang diterima dari form dengan aturan yang ditentukan. Jika validasi berhasil, menu baru akan dibuat menggunakan data dari request dan disimpan. Pengguna kemudian diarahkan kembali ke indeks menu dengan pesan sukses.

```
7  
8 class MenuController extends Controller  
9 {  
10    /**  
11     * Display a listing of the resource.  
12     */  
13    public function index()  
14    {  
15        $menus = Menu::all()  
16        return view('menus/index');  
17    }  
18  
19    /**  
20     * Show the form for creating a new resource.  
21     */  
22    public function create()  
23    {  
24        return view('menus/create');  
25    }  
26  
27    /**  
28     * Store a newly created resource in storage.  
29     */  
30    public function store(Request $request)  
31    {  
32        $menu = new Menu();  
33        $menu->name = $request->name;  
34        $menu->url = $request->url;  
35        $menu->parent_id = $request->parent_id;  
36        $menu->order = $request->order;  
37        $menu->status = $request->status;  
38  
39        $menu->save();  
40  
41        return redirect()->route('menus.index')  
42            ->with('success', 'Menu berhasil ditambahkan');  
43    }  
44  
45    /**  
46     * Display the specified resource.  
47     *  
48     * @param  \App\Menu  $menu  
49     * @return \Illuminate\Http\Response  
50     */  
51    public function show(Menu $menu)  
52    {  
53        return view('menus.show', compact('menu'));  
54    }  
55  
56    /**  
57     * Show the form for editing the specified resource.  
58     *  
59     * @param  \App\Menu  $menu  
60     * @return \Illuminate\Http\Response  
61     */  
62    public function edit(Menu $menu)  
63    {  
64        return view('menus/edit', compact('menu'));  
65    }  
66  
67    /**  
68     * Remove the specified resource from storage.  
69     *  
70     * @param  \App\Menu  $menu  
71     * @return \Illuminate\Http\Response  
72     */  
73    public function destroy(Menu $menu)  
74    {  
75        $menu->delete();  
76  
77        return redirect()->route('menus.index')  
78            ->with('success', 'Menu berhasil dihapus');
```

PERANCANGAN HALAMAN MENU

show(Menu \$menu)

Menampilkan detail dari menu tertentu. Ini menerima objek Menu dan meneruskannya ke tampilan 'menus/show'.

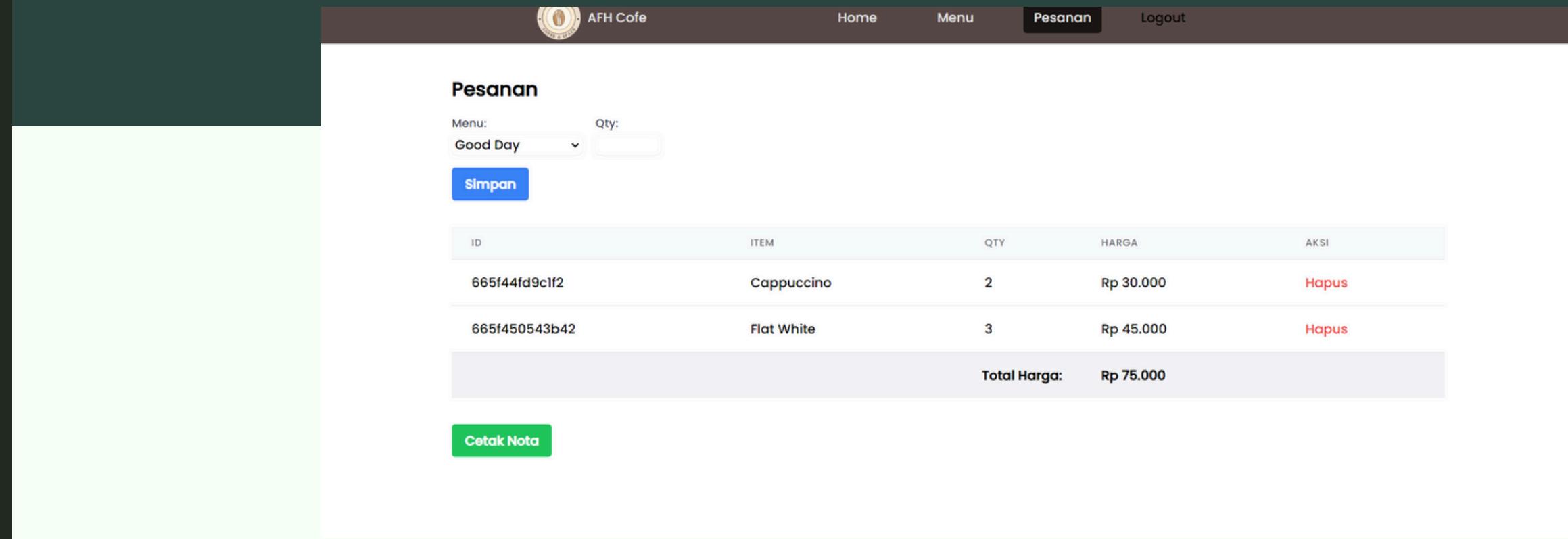
edit(Menu \$menu)

Menampilkan formulir untuk mengedit menu tertentu. Ini menerima objek Menu dan meneruskannya ke tampilan 'menus/edit'.

destroy(Menu \$menu)

Menghapus menu tertentu dari penyimpanan. Ini menerima objek Menu, menghapusnya, dan kemudian mengarahkan pengguna kembali ke indeks menu dengan pesan sukses.

Halaman Pesanan



The screenshot shows the 'Pesanan' (Order) section of the AFH Cafe website. At the top, there's a navigation bar with links for Home, Menu, Pesanan (selected), and Logout. Below the navigation is a form for adding items to the order:

Pesanan

Menu: Good Day Qty:

Simpan

ID	ITEM	QTY	HARGA	AKSI
665f44fd9c1f2	Cappuccino	2	Rp 30.000	Hapus
665f450543b42	Flat White	3	Rp 45.000	Hapus
Total Harga: Rp 75.000				

Cetak Nota

The table displays two items: Cappuccino (2 units at Rp 30.000 each) and Flat White (3 units at Rp 45.000 each). The total price is Rp 75.000. Each row has a red 'Hapus' (Delete) link under the AKSI column.

Pada halaman pesanan ini, Pengguna dapat menambahkan pesanan baru dengan memilih menu dari dropdown dan mengisi jumlah yang diinginkan. Setelah menekan tombol "Simpan", pesanan akan ditambahkan ke tabel di bawahnya. Tabel ini menampilkan detail pesanan seperti ID, item, jumlah, harga, dan aksi untuk menghapus pesanan. Total harga semua pesanan juga ditampilkan di bagian bawah tabel. Pengguna juga dapat mencetak nota pesanan dengan menekan tombol "Cetak Nota".

```
> app > Http > Controllers > 🐘 PesananController.php
```

```
class PesananController extends Controller
{
    public function index()
    {
        $pesanans = session()->get('pesanans');
        $menus = Menu::all();
        $totalHarga = collect($pesanans)
            ->sum(function ($item) {
                return $item['menu_id'] * $item['qty'];
            });

        return view('pesanans.index', [
            'pesanans' => $pesanans,
            'menus' => $menus,
            'totalHarga' => $totalHarga
        ]);
    }

    public function store(Request $request)
    {
        $request->validate([
            'id_item' => 'required|exists:menu,id',
            'qty' => 'required|integer'
        ]);

        $menu = Menu::findOrFail($request->input('id_item'));
        $total_price = $menu->harga * $request->input('qty');

        $pesanans = [
            'id' => uniqid(),
            'menu_id' => $menu->id,
            'qty' => $request->input('qty'),
            'harga' => $total_price
        ];
    }
}
```

PERANCANGAN HALAMAN PESANAN

index Method

Mengambil semua pesanan yang disimpan di sesi dan daftar menu dari database, serta menghitung total harga dari semua pesanan. Data ini kemudian dikirimkan ke tampilan pesanans.index untuk ditampilkan kepada pengguna.

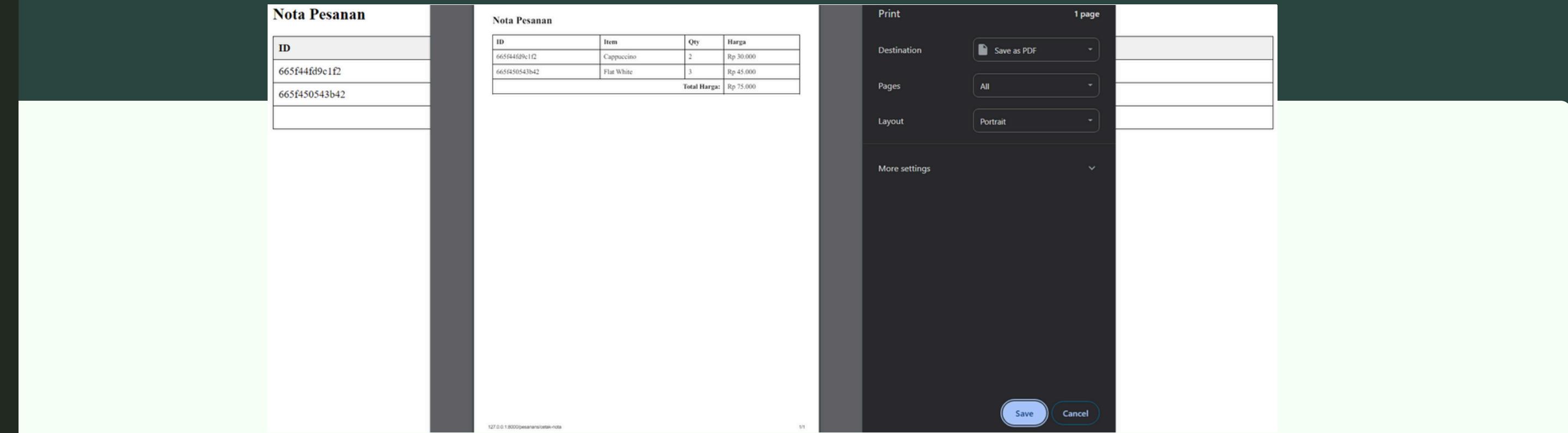
Store Method

Memvalidasi data pesanan yang diterima dari pengguna, menghitung total harga berdasarkan menu dan quantity yang dipilih, dan menyimpan pesanan tersebut ke dalam sesi. Pengguna kemudian diarahkan kembali ke halaman pesanan dengan pesan sukses.

CetakNota Method

Mengambil pesanan dari sesi dan menyimpannya ke dalam database. Setelah itu, data pesanan di sesi dihapus dan pesanan yang telah disimpan ditampilkan di halaman pesanans.nota untuk dicetak.

Cetak Nota



Setelah kasir melakukan pemesanan, mereka dapat mencetak nota sebagai bukti transaksi. Nota ini juga tersedia dalam format PDF untuk kemudahan aksesibilitas dan penyimpanan. Pengguna memiliki opsi untuk melihat daftar pesanan yang telah mereka inputkan, yang mencakup detail pesanan dan informasi transaksi.



TERIMAKASIH