

LAPORAN
DESKTOP APP DELTA ROMEO OUTDOOR
UNTUK MEMENUHI UAS

Mata kuliah : Pemrograman Berorientasi Objek
Dosen Pengampu : Mohamad Firdaus, M.Kom



Penyusun

Dimas Dwi Rianto (623C0004)

INSTITUT TEKNOLOGI DAN KESEHATAN MAHARDIKA
PROGRAM STUDI INFORMATIKA S1
2024

DAFTAR ISI

1. Deskripsi Aplikasi dan Tema

1.1 Deskripsi Singkat

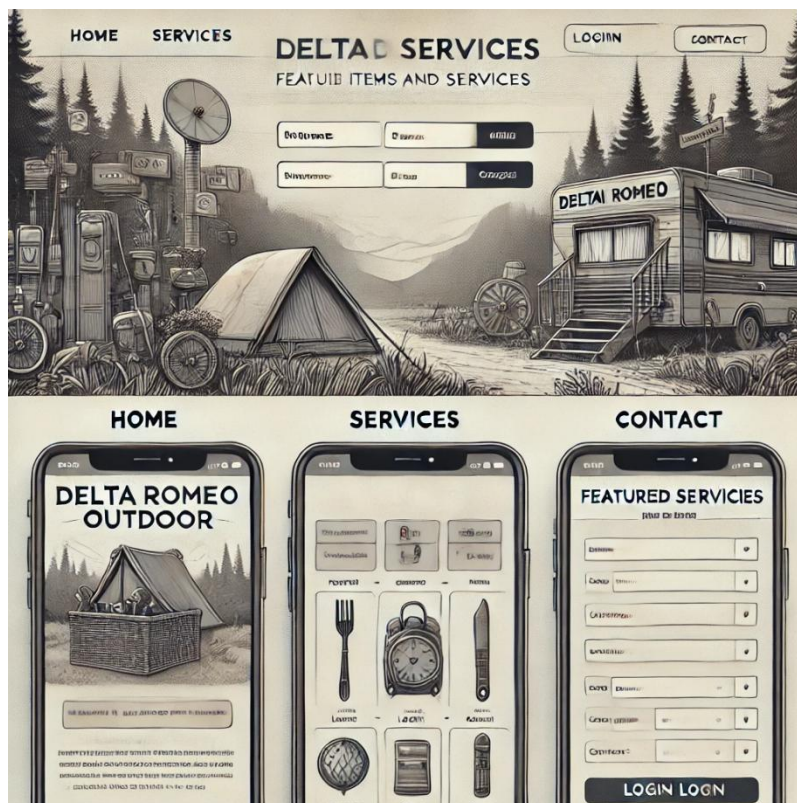
Desktop App yang dikembangkan adalah platform bernama DELTAROMELO OUTDOOR yang menyediakan layanan CRUD data barang yang hanya bisa dilakukan oleh admin, desktop app ini hanya bisa diakses oleh akun yang memiliki role sebagai admin, adapun fitur nya adalah menambah data barang baru, menghapus, mengedit.

1.2 Tema Aplikasi

Tema dari aplikasi ini adalah “Menyediakan Solusi Outdoor Praktis dan Terjangkau”. Desktop App ini dirancang untuk mendukung admin dari pada perusahaan ini sendiri untuk mempermudah pengelolaan data barang

2. Wireframe dan Desain Awal

2.1 Wireframe



2.2 Penjelasan Desain Antarmuka

Desain awal dirancang untuk mendukung pengalaman pengguna yang sederhana namun informatif. Dalam desain ini saya menggunakan beberapa kivy layouts yaitu seperti boxlayout, label, dan textinput

- Ada tabel untuk menampilkan informasi barang dari database
- Ada pilihan aksi untuk CRUD dari data barang tersebut

3. Implementasi Konsep Object-Oriented Programming (OOP)

Desktop app ini menggunakan Encapsulation, Inheritance, dan Polymorphism untuk mempermudah jalan nya apps dan mempermudah penggunaan aplikasi. Ada juga struktur yang digunakan pada desktop app kali ini ada Class untuk manage data dan Interface Controller untuk menghubungkan UI dengan logika ke database.

Pada proyek ini, pendekatan **Object-Oriented Programming (OOP)** digunakan untuk membangun aplikasi manajemen data dengan menggunakan **Kivy** sebagai framework utama. Berikut adalah penerapan konsep OOP dalam kode:

- **Inheritance (Pewarisan):** Kelas `InputForm` diturunkan dari kelas `BoxLayout`. Dengan pewarisan ini, `InputForm` dapat mewarisi atribut dan metode dari `BoxLayout`, seperti pengaturan tata letak dan elemen UI bawaan Kivy.
- **Encapsulation (Enkapsulasi):** Atribut-atribut seperti `mydb`, `mycursor`, dan `update_id` dideklarasikan dalam kelas `InputForm` untuk menyimpan koneksi database dan status aplikasi. Enkapsulasi memungkinkan pengelolaan atribut ini hanya dalam lingkup kelas.
- **Polymorphism (Polimorfisme):** Metode `show_data()` dalam kelas `InputForm` diimplementasikan untuk menangani pengambilan data dari elemen UI secara fleksibel. Metode ini dapat diadaptasi untuk berbagai skenario input tanpa mengubah logika inti.
- **Integrasi dengan Database:** Aplikasi ini menggunakan `mysql.connector` untuk menghubungkan ke database MySQL. Metode dalam kelas `InputForm` seperti `show_data()` mengilustrasikan bagaimana data diambil dari form, divalidasi, dan dikirim ke database untuk operasi CRUD.

4. Desain Antarmuka

Desain Antarmuka Aplikasi

- **Wireframe:** Wireframe aplikasi terdiri dari beberapa elemen:
 - Kolom input untuk ID Barang, Nama Barang, Harga, dan Stok.
 - Tombol untuk menyimpan, memperbarui, atau menghapus data.
 - Tabel atau daftar untuk menampilkan data yang diambil dari database.
- **Implementasi UI dengan Kivy:** Implementasi antarmuka pengguna dilakukan menggunakan file deltaromeo.kv. Berikut adalah elemen penting:
 - **Struktur Layout:** Menggunakan BoxLayout dengan orientasi vertikal dan horizontal untuk mengatur tata letak elemen seperti label, input teks, dan tombol.
 - **Gaya Visual:** Warna latar belakang diterapkan dengan Canvas dan nilai RGBA untuk memberikan tampilan estetik.
 - **Komponen Form:** Elemen seperti Label dan TextInput digunakan untuk menerima input dari pengguna, sementara gambar (logo.png) menambahkan branding visual.

5. Tantangan yang Dihadapi

- **Tantangan 1: Validasi Input Pengguna** Beberapa form input membutuhkan validasi agar tidak kosong atau salah format. Contoh:
 - Memastikan input harga adalah angka valid.
 - Mencegah penyimpanan data jika form kosong.

Solusi: Validasi dilakukan dalam metode seperti `show_data()`. Jika input tidak valid, aplikasi menampilkan pesan error menggunakan popup.

- **Tantangan 2: Koneksi Database** Terkadang, koneksi ke database gagal karena konfigurasi server lokal atau kredensial salah.

Solusi: Menggunakan blok `try-except` untuk menangani error koneksi database, sehingga aplikasi tetap responsif meskipun terjadi kesalahan.

- **Tantangan 3: Desain Responsif** Antarmuka harus dapat menyesuaikan ukuran layar agar nyaman digunakan pada perangkat berbeda.

Solusi: Properti seperti `size_hint` dan `spacing` digunakan dalam elemen layout untuk memastikan antarmuka bersifat fleksibel

6. Kesimpulan dan Pelajaran yang Dipetik

Tantangan terbesar adalah memastikan keamanan data pengguna dan koneksi database ke website, pengalaman ini memberikan saya wawasan dalam membangun aplikasi berbasis web termasuk cara memadukan front-end dengan logika back-end.

Kesimpulan yang bisa saya buat adalah proyek pengembangan website DELTAROMEO OUTDOOR memberikan Solusi praktis bagi para pecinta outdoor untuk menyewa alat camping secara online. Meskipun sederhana, aplikasi ini sudah mencakup fitur utama seperti manajemen barang, autentikasi pengguna, formulir pemesanan.

Meskipun ini sulit tapi ketika sudah dilakukan dan kita ada kemauan untuk mencoba, semua akan terasa ringan dan bisa dilakukan.

Proyek ini memberikan pengalaman mendalam dalam mengembangkan aplikasi GUI berbasis Kivy dengan pendekatan OOP. Beberapa pelajaran penting yang didapatkan meliputi:

- **Penguasaan Framework Kivy:** Proyek ini memperkuat pemahaman tentang bagaimana elemen UI didefinisikan dalam file `.kv` dan diintegrasikan dengan logika Python.
- **Manajemen Database:** Koneksi dan operasi CRUD dengan MySQL memberikan wawasan tentang bagaimana aplikasi desktop dapat berinteraksi dengan basis data secara efisien.
- **Penerapan Prinsip OOP:** Penggunaan OOP memungkinkan kode menjadi lebih modular, dapat dipelihara, dan mudah diperluas untuk fitur tambahan di masa depan.

Melalui proyek ini, berbagai tantangan berhasil diatasi dengan solusi kreatif, memberikan dasar yang kuat untuk pengembangan aplikasi yang lebih kompleks di masa mendatang.