

LAMPIRAN

Lampiran 1 Surat Kesiediaan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Muhamad Ilham Nur Habibie Wibowo
NIM : 20090069
Program Studi : D IV Teknik Informatika

Pihak Kedua

Nama : Dega Surono Wibowo, S.T., M.Kom.
Status : Dosen Tetap
NIDN : 0607108202
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata / III/C

Pada hari ini Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan melakukan bimbingan 1x dalam seminggu atau setidaknya 3x bimbingan dalam 1 bulan (dengan progres). Apabila saya tidak memenuhi syarat tersebut, maka saya tidak berhak meminta surat rekomendasi mengikuti sidang Skripsi. Dan saya berjanji akan memenuhi persyaratan tersebut dan menyelesaikan penelitian (produk dan laporan) sesuai jadwal penelitian (tepat waktu).

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama



Muhamad Ilham Nur Habibie Wibowo

Pihak Kedua



Dega Surono Wibowo, S.T., M.Kom.
NIPY: 06.014.183

Mengetahui

Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.

NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Muhamad Ilham Nur Habibie Wibowo
NIM : 20090069
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Status : Dosen Tetap
NIDN : 0623108801
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk. I/III-d

Pada hari ini Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama berjanji akan menyelesaikan Skripsi dalam waktu maksimal 3 bulan (awal bulan Juni 2024). Jika syarat tersebut tidak terpenuhi, maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama



Muhamad Ilham Nur Habibie Wibowo

Pihak Kedua



Muhammad Fikri Hidayattullah, S.T., M.Kom.

Mengetahui

Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.

NIPY. 09.015.225

Lampiran 2 Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Muhamad Ilham Nur Habibie Wibowo
Kewarganegaraan : Indonesia
Alamat : Jl. Dr. Setia Budi No.28 RT.01/RW.01, Kembang Baru,
Kecamatan Brebes, Kabupaten Brebes, Jawa Tengah 52212
2. Nama : Dega Suruno Wibowo, S.T., M.kom.
Kewarganegaraan : Indonesia
Alamat : Perum Sapphire Regency Blok H No.1, RT.004/RW.001,
Kelurahan Pulosari, Kecamatan Brebes, Jawa Tengah 52213
3. Nama : Muhammad Fikri Hidayattullah. S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68, Randugunting, Kecamatan Tegal Selatan,
Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:

Berupa : Program Komputer
Berjudul : Barberbot: Aplikasi Layanan Booking Barbershop Berbasis
Android

Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);

- Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
 3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 24 Juli 2024



(Muhamad Ilham Nur Habibie Wibowo)
Pemegang Hak Cipta *

(Dega Surono Wibowo, S.T., M.Kom.)
Pemegang Hak Cipta *

(Muhammad Fikri Hidayattullah, S.T., M.Kom.)
Pemegang Hak Cipta *

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3 Surat Pengalihan Hak Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Muhamad Ilham Nur Habibie Wibowo
Kewarganegaraan : Indonesia
Alamat : Jl. Dr. Setia Budi No.28 RT.01/RW.01, Kembang Baru,
Kecamatan Brebes, Kabupaten Brebes, Jawa Tengah 52212
2. Nama : Dega Suruno Wibowo, S.T., M.kom.
Kewarganegaraan : Indonesia
Alamat : Perum Sapphire Regency Blok H No.1, RT.004/RW.001,
Kelurahan Pulosari, Kecamatan Brebes, Jawa Tengah 52213
3. Nama : Muhammad Fikri Hidayattullah. S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68, Randugunting, Kecamatan Tegal Selatan,
Kota Tegal

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Politeknik Harapan Bersama
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Barberbot: Aplikasi Layanan Booking Barbershop Berbasis Android", untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.


Tegal, 24 Juli 2024

Pemegang Hak Cipta


(Dr. Aldi Budi Riyanta, S.Si., M.T.)

Pencipta


(Muhamad Ilham Nur Habibie Wibowo)


(Dega Surono Wibowo, S.T., M.Kom.)


(Muhammad Fikri Hidayattullah, S.T., M.Kom.)

Lampiran 4 Manual *Book*



MANUAL BOOK

APLIKASI BOOKING BARBERSHOP : BARBERBOT APLIKASI PEMESANAN
LAYANAN BARBERSHOP



Penulis :

MUHAMAD ILHAM NUR HABIBIE WIBOWO

1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Barberbot adalah aplikasi layanan booking barbershop yang inovatif, dirancang untuk mempermudah pengguna dalam mengatur jadwal kunjungan ke barbershop dan memilih gaya rambut yang sesuai dengan preferensi dan bentuk wajah mereka. Dengan fitur-fitur yang canggih dan antarmuka yang user-friendly, Barberbot bertujuan untuk meningkatkan efisiensi dan kepuasan pengguna dalam memperoleh layanan barbershop.

1.2. Deskripsi Umum

1.2.1 Deskripsi Umum Aplikasi

Barberbot adalah aplikasi mobile untuk pelayanan barbershop yang menggunakan teknologi deep learning untuk mendeteksi bentuk wajah user dan memberikan rekomendasi model potongan rambut yang sesuai dengan bentuk wajah. Aplikasi mendapat inputan gambar dari galeri atau menggunakan kamera secara langsung untuk melakukan analisis bentuk wajah. Selain itu, pengguna juga dapat melakukan order dan booking langsung melalui aplikasi ini.

1.2.2. Deskripsi Umum Kebutuhan Aplikasi

Barberbot adalah aplikasi mobile untuk pelayanan barbershop yang dapat mendeteksi bentuk wajah menggunakan teknologi deep learning dan memberikan rekomendasi model potongan rambut. Aplikasi ini menggunakan flask untuk back-end, dan flutter untuk front-end, serta MySQL untuk penyimpanan data. Aplikasi ini dirancang agar pengguna dapat dengan mudah memesan layanan barbershop sesuai hari dan waktu yang diinginkan.

1.3 Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan yang jelas dan komprehensif kepada pengguna dalam menggunakan aplikasi Barberbot, serta untuk menjelaskan teknologi dan integrasi yang digunakan dalam pengembangan aplikasi ini. Dokumen ini berisikan informasi sebagai berikut:

1. BAB I
Memperkenalkan konsep dan tujuan aplikasi Barberbot, serta menjelaskan teknologi deep learning yang digunakan untuk mendeteksi bentuk wajah.
2. BAB II
Menyajikan detail mengenai antarmuka pengguna aplikasi, termasuk fitur-fitur utama seperti order, booking, upload gambar, dan hasil rekomendasi model potongan rambut.
3. BAB III
Fokus kepada teknologi yang digunakan dalam pengembangan aplikasi, mencakup flask sebagai back-end, flutter sebagai front-end, dan MySQL sebagai manajemen data.

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak

- Flask (sebagai back-end)
- Flutter (sebagai front-end)
- MySQL (sebagai basis data)
- Python (untuk pengembangan flask dan model deep learning)
- TensorFlow (untuk mengembangkan model deep learning)
- Library atau Framework untuk deep learning seperti keras

2.2. Perangkat Keras

- Komputer atau Laptop untuk mengembangkan aplikasi, yang memenuhi spesifikasi minimum untuk menjalankan Flask, Python, dan perangkat lunak lainnya.
- Kamera atau Webcam yang dapat terintegrasi dengan aplikasi untuk pengambilan gambar wajah.
- Peralatan jaringan yang memastikan koneksi ke internet untuk pengambilan data model.

3. MENU DAN CARA PENGGUNAAN

3.1. Struktur Menu

Berikut adalah struktur menu yang ada di aplikasi Barberbot.

a. Menu Level Barberman

- Login
- Orderan

b. Menu Level Owner

- Login
- Laporan

c. Menu Level User

- Login
- Register
- Order
- Booking
- Artikel
- Chatbot
- Profile

3.2. Penggunaan Aplikasi

Berikut adalah alur penggunaan aplikasi Barberbot beserta penjelasan masing-masing level.

1. Pengguna Aplikasi Level Barberman

Di bagian ini admin dapat mengetahui jika ada order atau booking yang masuk.

a) Login Barberman

Pada halaman ini, terdapat form login admin Dimana harus memasukan username dan password yang sesuai. Tampilan form login admin dapat di lihat pada gambar 3.1 di bawah ini.



Gambar 3. 1 *Login*

b) Halaman Orderan *Barberman*

Pada halaman ini menampilkan orderan yang terdiri dari filter tanggal, *customers* order, dan laporan. Kemudian admin juga dapat memberi aksi untuk orderan pelanggan. Halaman *home* admin dapat dilihat pada gambar 3.2.

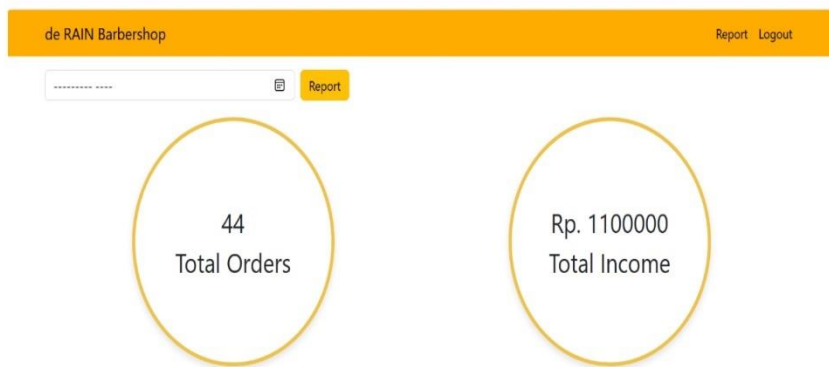
de RAIN Barbershop						Report	Logout				
dd/mm/yyyy		all	Filter								
Order				Booking							
No	Nama	Face	Image	Date	Type	No	Nama	Face	Image	Date	Type
1	muhamad Ilham Nur Habibie Wibowo			2024- 07-23	order	1	muhamad Ilham Nur Habibie Wibowo			2024- 07-23 02:43:23	orde
2	muhamad Ilham Nur Habibie Wibowo			2024- 07-23	bookin	2	muhamad Ilham Nur Habibie Wibowo			2024- 07-23 11:00:00	bool
3	muhamad Ilham Nur			2024- 07-23	bookin	3	muhamad Ilham Nur			2024- 07-23	bool

Gambar 3. 2 Orderan *Barberman*

2. Pengguna Aplikasi Level Owner

a) Laporan Owner

Pada halaman ini menampilkan laporan yang terdiri dari rekap daftar pelanggan dan penghasilan dalam 1 bulan. Halaman laporan dapat dilihat pada gambar 3.3.



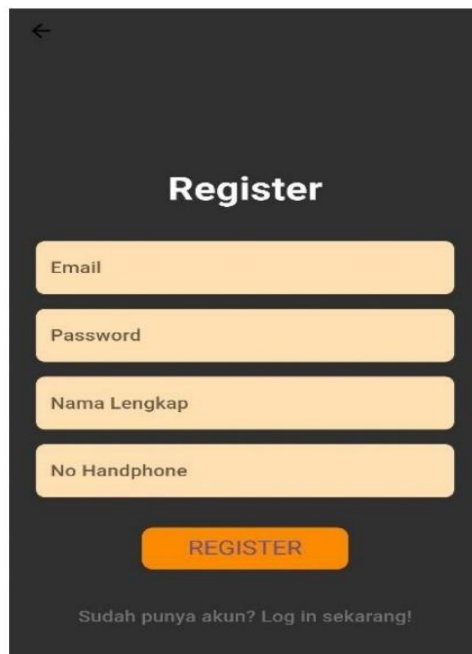
Gambar 3. 3 Laporan Owner

2. Pengguna Aplikasi Level Customer

Di bagian ini, pelanggan dapat mengetahui bentuk wajah dan melihat hasil rekomendasi model potongan rambut.

a. Halaman *Register Customer*

Jika *user* ingin melakukan login, tetapi *user* belum mendaftar atau belum mempunyai akun, maka *user* tidak akan bisa *login*. Karena itu, *user* yang belum mempunyai akun diharuskan untuk mendaftar akun terlebih dahulu di halaman *register*. Pada halaman *register*, terdapat *form register* yang di mana *user* harus memasukan *email,password,username*, dan *no.hp*. Jika *user* sudah berhasil daftar, *user* akan diarahkan ke halaman *home*. Tampilan *form register* dapat di lihat pada gambar 3.4.



The image shows a mobile application registration screen with a dark background. At the top left is a back arrow. The title 'Register' is centered in white. Below it are four light blue input fields with labels: 'Email', 'Password', 'Nama Lengkap', and 'No Handphone'. A blue button with the text 'REGISTER' is positioned below the fields. At the bottom, there is a link that says 'Sudah punya akun? Log in sekarang!'.

Gambar 3. 4 *Register*

b. Halaman *Login Customer*

Pada halaman ini, terdapat form login Dimana *user* harus memasukan username dan password yang sesuai Ketika mendaftar akun. Tampilan form *login* dapat di lihat pada gambar 3.5.

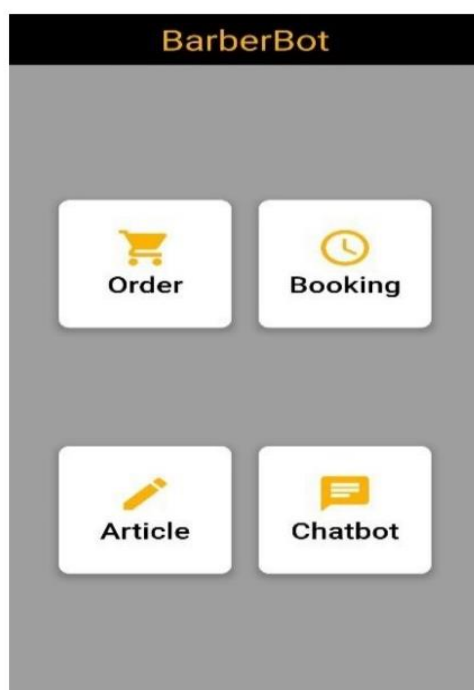


Gambar 3.5 *Login*

c. Halaman *Home Customer*

Halaman utama aplikasi Barberbot menampilkan empat pilihan fitur utama yang mudah diakses. Desain minimalis memfokuskan pada fitur-fitur yang tersedia, membuatnya mudah digunakan dan dipahami. Pada halaman ini, Anda dapat memilih untuk memesan, membaca artikel tentang perawatan rambut, memesan produk, atau berinteraksi dengan chatbot

untuk mendapatkan bantuan. Tampilan *Home* dapat di lihat pada gambar 3.6.



Gambar 3. 6 *Home*

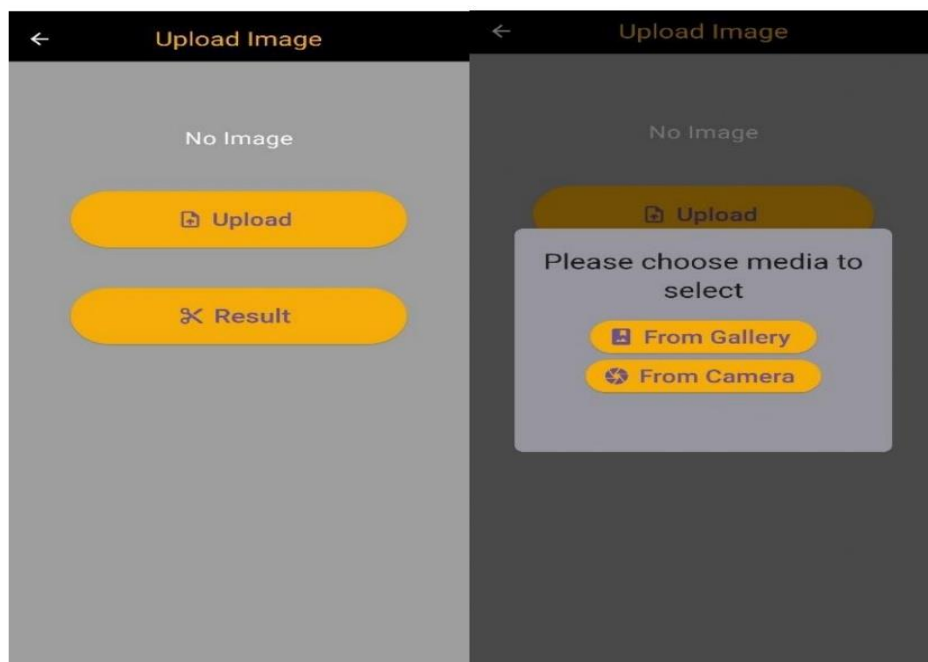
d. Order

Pada halaman Order berisikan tampilan antarmuka yang dibangun untuk memandu pengguna melalui proses pemesanan layanan barbershop. Ada beberapa tahap dalam *interface* order.

1) *Upload Image*

Pada *Upload Image* pengguna diminta untuk menginput gambar wajah melalui *device*. Setelah pengguna klik tombol "Upload" maka akan keluar sebuah *pop up* pilihan untuk meng upload dari kamera atau galeri. Jika

sudah menginput gambar dan klik tombol “Result” menggunakan di arahan ke antarmuka result. Tampilan *upload image* dapat di lihat pada gambar 3.7.



Gambar 3.7 *Upload Image*

2) *Result*

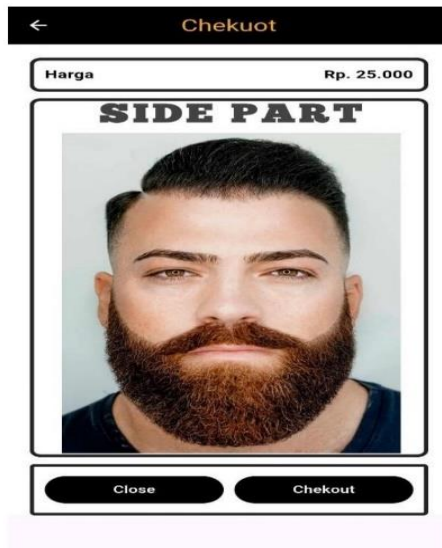
Pada *Result* antarmuka menampilkan hasil deteksi bentuk wajah dan menampilkan rekomendasi gaya rambut. Terdapat 4 pilihan gaya rambut yang dapat dipilih oleh user. Jika user meng klik salah satu gaya rambut, maka pengguna akan diarahkan ke antarmuka *checkout*. Tampilan halaman result dapat dilihat pada gambar 3.8.



Gambar 3. 8 Result

3) Chekout Order

Pada halaman *Chekout* antarmuka menampilkan halaman untuk menyelesaikan proses *chekout*.



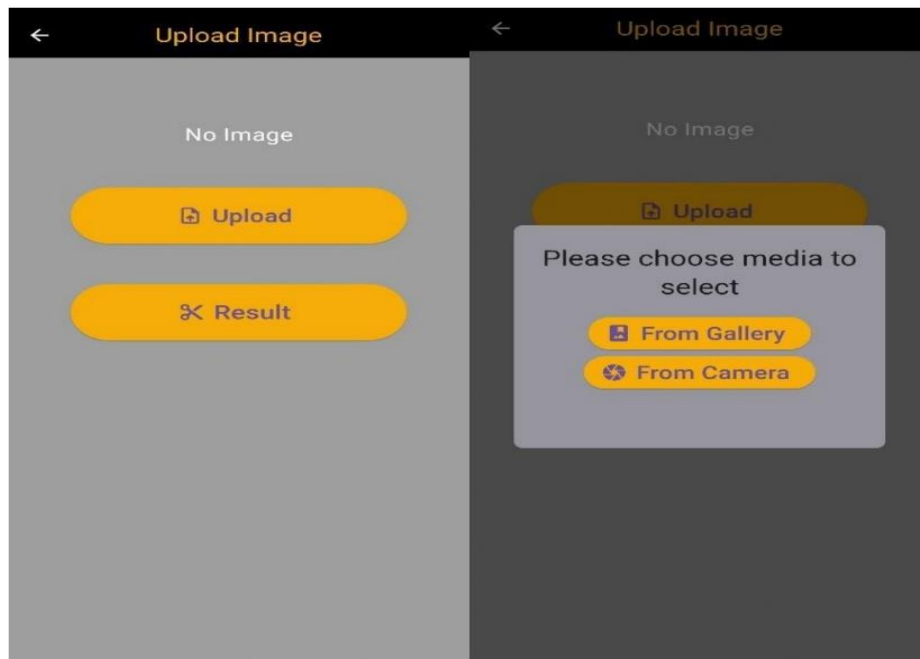
Gambar 3. 9 Chekout

e. Booking

Pada halaman *booking* berisikan tampilan antarmuka yang dibangun untuk memandu pengguna melalui proses pemesanan layanan barbershop. Ada beberapa tahap dalam *interface booking*.

1) *Upload Image*

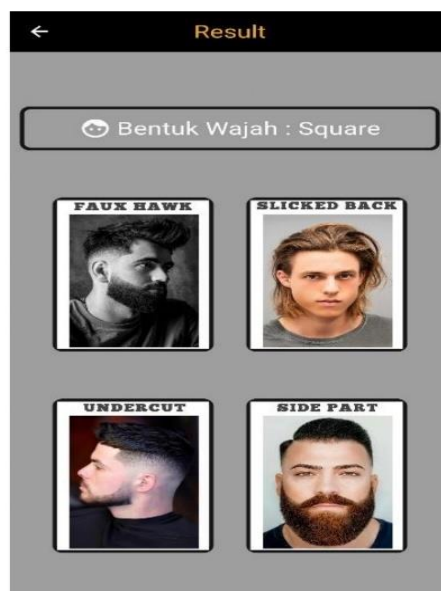
Pada *Upload Image* pengguna diminta untuk menginput gambar wajah melalui *device*. Setelah pengguna klik tombol “Upload” maka akan keluar sebuah *pop up* pilihan untuk meng upload dari kamera atau galeri. Jika sudah menginput gambar dan klik tombol “Result” pengguna akan di arahkan ke antarmuka result. Tampilan *upload image* dapat di lihat pada gambar 3.10.



Gambar 3. 10 *Upload Image*

2) *Result*

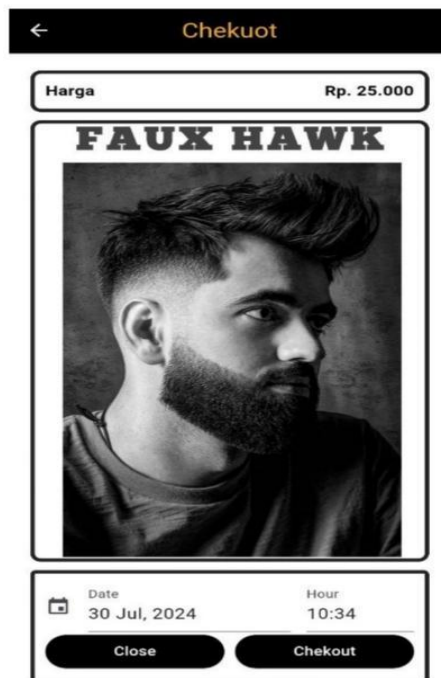
Pada *Result* antarmuka menampilkan hasil deteksi bentuk wajah dan menampilkan rekomendasi gaya rambut. Terdapat 4 pilihan gaya rambut yang dapat dipilih oleh user. Jika user meng klik salah satu gaya rambut, maka pengguna akan diarahkan ke antarmuka *checkout*. Tampilan halaman result dapat dilihat pada gambar 3.11.



Gambar 3. 11 *Result*

3) *Checkout Order*

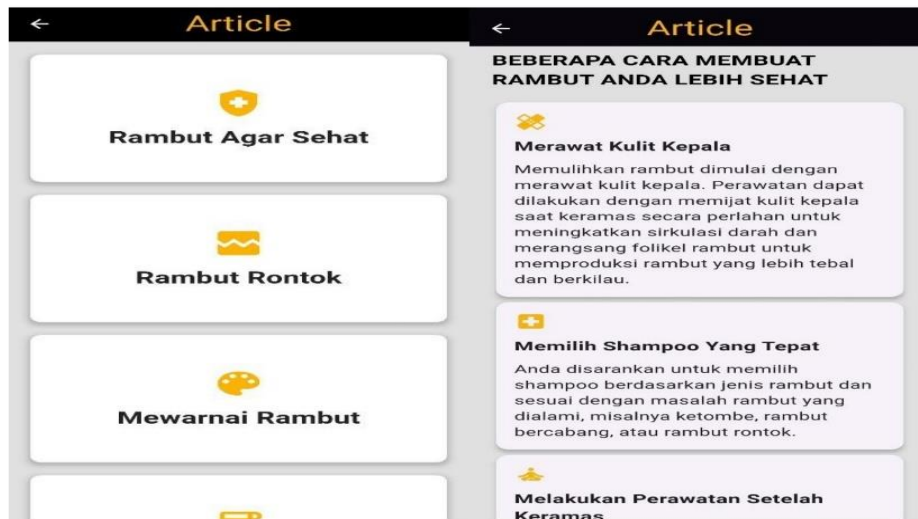
Pada *Checkout Booking* antarmuka menampilkan halaman untuk menyelesaikan proses *checkout*. Sebelum melakukan *checkout* *user* dapat memilih tanggal dan waktu untuk melakukan *booking*. Tampilan halaman *checkout* dapat dilihat pada gambar 3.12.



Gambar 3. 12 *Chekout*

f. Halaman Artikel

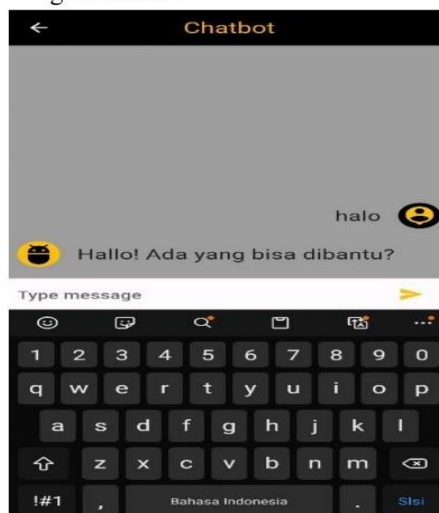
Interface articlel menampilkan konten informasi seputar rambut. Pengguna dapat memilih *article* dan melihat detail *article*. Tampilan pada halaman artikel dapat di lihat pada gambar 3.13.



Gambar 3. 13 Artikel

g. Halaman Chatbot

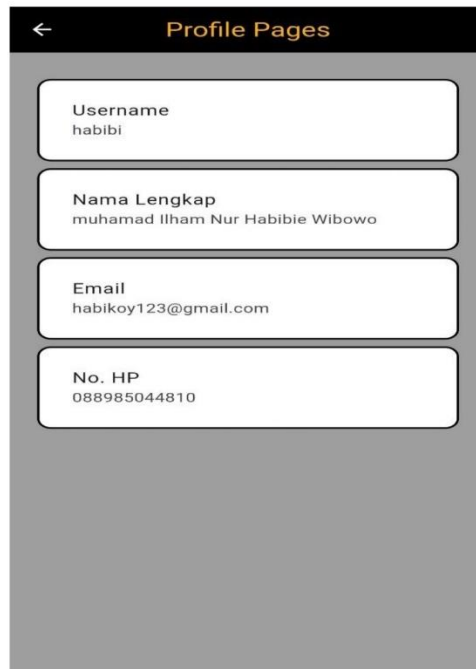
Interface Chatbot adalah fitur interaktif yang memungkinkan pengguna untuk berkomunikasi dengan sistem secara langsung dan mendapatkan jawaban atas pertanyaan-pertanyaan seputar aplikasi. Tampilan pada halaman chatbot dapat dilihat pada gambar 3.14.



Gambar 3. 14 Chatbot

h. Profile

Interface Profile dirancang untuk menampilkan informasi user. Setelah pengguna membuka halaman profile, mereka akan melihat data seperti username, nama lengkap, email, dan nomer handphone. Tampilan pada halaman histori dapat dilihat pada gambar 3.15.



Gambar 3. 15 Profile

Lampiran 5 Dokumen Teknikal



TECHNICAL DOCUMENT

APLIKASI BOOKING BARBERSHOP : BARBERBOT APLIKASI PEMESANAN
LAYANAN BARBERSHOP



Penulis :
MUHAMAD ILHAM NUR HABIBIE WIBOWO

Profil

Barberbot adalah aplikasi layanan booking barbershop yang inovatif, dirancang untuk mempermudah pengguna dalam mengatur jadwal kunjungan ke barbershop dan memilih gaya rambut yang sesuai dengan preferensi dan bentuk wajah mereka. Dengan fitur-fitur yang canggih dan antarmuka yang user-friendly, Barberbot bertujuan untuk meningkatkan efisiensi dan kepuasan pengguna dalam memperoleh layanan barbershop.

Latar Belakang

Di era digital saat ini, permintaan akan layanan yang praktis dan efisien semakin meningkat, termasuk dalam industri fashion dan kecantikan. Salah satu contohnya adalah layanan barbershop, yang memainkan peran penting dalam meningkatkan penampilan dan kepercayaan diri seseorang. Mengunjungi barbershop seringkali membutuhkan waktu dan perencanaan, terutama bagi mereka yang memiliki jadwal yang padat dan sibuk. Oleh karena itu, ada kebutuhan yang mendesak untuk mengembangkan aplikasi yang dapat membantu mempermudah proses pemesanan layanan barbershop. Aplikasi Barberbot hadir sebagai solusi untuk mengatasi tantangan ini. Dengan Barberbot, pengguna dapat dengan mudah menemukan dan memesan layanan barbershop sesuai dengan jadwal mereka. Tidak hanya itu, aplikasi ini juga menawarkan fitur unik berupa rekomendasi gaya rambut yang disesuaikan dengan preferensi dan bentuk wajah pengguna. Hal ini tidak hanya mempermudah pengguna dalam memilih gaya rambut yang sesuai, tetapi juga meningkatkan kepuasan dan kepercayaan diri mereka. Dalam pengembangannya, Barberbot dirancang untuk memberikan pengalaman pengguna yang optimal dengan antarmuka yang sederhana dan intuitif. Aplikasi ini mengintegrasikan teknologi terkini untuk analisis bentuk wajah dan preferensi gaya rambut, sehingga menghasilkan rekomendasi yang akurat dan personal. Dengan demikian, Barberbot diharapkan dapat menjadi alat yang berguna dan efisien dalam mengoptimalkan layanan barbershop, mempermudah pengguna dalam mengatur jadwal, dan membantu mereka menemukan gaya rambut yang paling sesuai.

Manfaat

1. Pengguna dapat dengan mudah memilih barbershop, serta menentukan tanggal dan waktu kunjungan yang sesuai dengan jadwal mereka melalui aplikasi. Ini mengurangi kebutuhan untuk menelepon atau datang langsung ke barbershop untuk membuat janji.
2. Dengan fitur pemesanan yang cepat dan efisien, pengguna dapat menghemat waktu yang biasanya dihabiskan untuk mencari barbershop dan membuat janji. Pengguna bisa langsung melihat jadwal ketersediaan dan melakukan booking dalam hitungan menit.
3. Aplikasi ini menggunakan teknologi analisis bentuk wajah dan preferensi gaya rambut untuk memberikan rekomendasi gaya rambut yang sesuai. Ini membantu pengguna menemukan gaya yang paling cocok dengan penampilan mereka, meningkatkan kepercayaan diri.

Arsitektur Aplikasi

- Deep Learning : Menggunakan Convolutional Neural Network (CNN) untuk menganalisis bentuk wajah (Square, Oval, Oblong, Round, dan Heart).
- Framework: Aplikasi ini dibangun menggunakan Flask untuk back-end dan Flutter untuk front-end. MySQL digunakan sebagai database untuk menyimpan data.
- Webcam & Upload: Memungkinkan pengguna untuk mengambil foto langsung dari kamera atau mengunggah foto yang sudah ada.

Spesifikasi Teknis

1. Perangkat Keras
 - a) Server
 - CPU: Minimal Intel Xeon atau setara
 - RAM: Minimal 16 GB
 - Penyimpanan: Minimal 500 GB SSD
 - Koneksi Internet: Stabil dengan bandwidth tinggi

b) Pengguna

- PC/Laptop dengan spesifikasi dasar (CPU Intel i5 atau setara, RAM 8 GB)
- Koneksi Internet yang stabil
- Kamera web (untuk deteksi langsung)

2. Perangkat Lunak

Perangkat lunak yang digunakan untuk membuat aplikasi :

- Bahasa Pemrograman: Flutter
- Database: MySQL
- Library dan Framework: TensorFlow, Keras, OpenCV (untuk pemrosesan gambar)

3. Source Code

Beberapa source code dalam membangun website Barberbot dijelaskan sebagai berikut:

a. Login.dart

- Controller untuk Email dan Password: Membuat TextEditingController untuk menyimpan input email dan password pengguna.
- Form HTML dengan Method POST: Di Flutter, kita menggunakan widget TextField untuk mengumpulkan input pengguna dan widget ElevatedButton untuk mengirimkan data.
- Input c_email untuk email pengguna dan c_pass untuk password: Menggunakan TextField dengan controller untuk email dan password.
- Tombol Login: ElevatedButton yang ketika diklik akan mengirimkan form.

```

login.dart M X
lib > widgets > login.dart > LoginPage > build
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:mobile/controller/login.dart';
4 import 'package:mobile/widgets/register.dart'; // Add this line
5
6 LoginController controller = Get.put(LoginController());
7
8 class LoginPage extends StatelessWidget {
9   const LoginPage({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         backgroundColor: Colors.transparent,
16         elevation: 0,
17         leading: IconButton(
18           icon: const Icon(Icons.arrow_back, color: Colors.black),
19           onPressed: () {
20             Navigator.pop(context);
21           },
22         ), // IconButton
23       ), // AppBar
24       body: Center(
25         child: Container(
26           width: double.infinity,
27           padding: const EdgeInsets.symmetric(horizontal: 24.0),
28           child: Box() {
29             return ListView(
30               children: [
31                 Text(
32                   'Log In',
33                   style: TextStyle(
34                     fontSize: 32,
35                     fontWeight: FontWeight.bold,
36                     color: Colors.white,
37                   ), // TextStyle
38                 ), // Text
39                 const SizedBox(height: 32),
40                 TextFormField(
41                   keyboardType: TextInputType.emailAddress,
42                   decoration: InputDecoration(
43                     hintText: 'Email',
44                     filled: true,
45                     fillColor: Colors.orange[100],
46                     border: OutlineInputBorder(
47                       borderRadius: BorderRadius.circular(8.0),
48                       borderSide: BorderSide.none,
49                     ), // OutlineInputBorder
50                   ), // InputDecoration
51                   controller: controller.username,
52                 ), // TextFormField
53                 const SizedBox(height: 16),
54                 TextFormField(
55                   obscureText: true,
56                   decoration: InputDecoration(
57                     hintText: 'Password',
58                     filled: true,
59                     fillColor: Colors.orange[100],
60                     border: OutlineInputBorder(
61                       borderRadius: BorderRadius.circular(8.0),
62                       borderSide: BorderSide.none,
63                     ), // OutlineInputBorder
64                   ), // InputDecoration
65                   controller: controller.password,
66                 ), // TextFormField
67                 const SizedBox(height: 8),
68                 Align(
69                   alignment: Alignment.centerRight,
70                   child: TextButton(
71                     onPressed: () {},
72                     child: const Text('Lupa kata sandi?'),
73                   ), // TextButton
74                 ), // Align
75                 const SizedBox(height: 32),
76                 ElevatedButton(
77                   onPressed: () {
78                     if (!controller.isLoading.value) {
79                       controller.onSubmit();
80                     }
81                   },
82                   style: ElevatedButton.styleFrom(
83                     backgroundColor: Colors.orange[600],
84                     padding: const EdgeInsets.symmetric(
85                       horizontal: 40.0, vertical: 10.0), // EdgeInsets.symmetric
86                   ),
87                   text: const Text('LOG IN',
88                     style: TextStyle(fontSize: 20),
89                     shape: RoundedRectangleBorder(
90                       borderRadius: BorderRadius.circular(10.0)), // RoundedRectangleBorder
91                   ),
92                   child: controller.isLoading.value
93                     ? SizedBox(
94                       height: 20,
95                       width: 20,
96                       child: CircularProgressIndicator(
97                         color: Colors.white,
98                       ), // CircularProgressIndicator
99                     ) : const Text('LOG IN'),
100                 ), // ElevatedButton
101                 const SizedBox(height: 26),
102                 TextButton(
103                   onPressed: () {},
104                   child: const Text(''),
105                 ), // TextButton
106                 const SizedBox(height: 16),
107                 TextButton(
108                   onPressed: () {
109                     Get.off(RegisterPage());
110                   },
111                   child: Text(
112                     'Belum punya akun? Daftar sekarang!',
113                     style: TextStyle(

```

- Validasi Input: Validasi input untuk mencegah serangan injeksi SQL dan serangan XSS.
- Pengelolaan File: Mengatur jenis file yang diunggah dan membatasi ukuran file untuk keamanan aplikasi.

```

27 from klasifikasi import cnn_model
28 from nltk.stem import WordNetLemmatizer
29 from nltk.tokenize import word_tokenize
30 from PIL import Image as im
31
32 app = Flask(__name__)
33 CORS(app)
34 app.secret_key = "check"
35
36 app.secret_key = "secret key"
37 app.config['MAX_CONTENT_LENGTH'] = 4024 * 4024
38 app.config['UPLOAD_EXTENSIONS'] = ['.jpg', '.png', '.jpeg', '.JPG', '.JPEG']
39 app.config['UPLOAD_PATH'] = './static/images/uploads/'
40 app.config['REKOMENDASI_PATH'] = './static/images/recommendation/'
41
42 ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg', 'gif'])
43
44 app.config['MYSQL_HOST'] = 'localhost'
45 app.config['MYSQL_USER'] = 'root'
46 app.config['MYSQL_PASSWORD'] = ''
47 app.config['MYSQL_DB'] = 'hair'
48
49 mysql = MySQL(app)
50
51 model_cnn = cnn_model()
52
53 lemmatizer = WordNetLemmatizer()
54
55 with open('chatbot/chatbot.json') as content:

```

d. Menjalankan Aplikasi

Menjalankan flask tersebut dalam terminal menggunakan perintah “python app.py” jika berhasil mengerun maka akan muncul link port seperti di bawah ini:

```

if __name__ == "__main__":
    model.load_weights('./chatbot/chatbotmodel.h5')
    model_cnn.load_weights('./model.h5')
    app.run(host="192.168.18.70", debug=True, port=33, threaded=False)

```


- Aplikasi Flask diinisialisasi dengan menggunakan Flask(__name__) dan menggunakan CORS(app) untuk mengizinkan akses dari berbagai domain.
- Konfigurasi app.secret_key digunakan untuk keamanan session dan app.config untuk menentukan konfigurasi upload gambar dan koneksi MySQL.
- Pengaturan koneksi MySQL dilakukan dengan konfigurasi app.config['MYSQL_*'] untuk menghubungkan Flask dengan database MySQL.
- Model neural network untuk klasifikasi teks menggunakan Keras Sequential API, dengan lapisan-lapisan seperti Dense, Dropout, dan optimizer SGD.
- Penggunaan NLTK untuk tokenisasi kata dan lematisasi dalam pemrosesan teks, memungkinkan aplikasi untuk memahami maksud dari pola kata-kata yang diterima.
- Aplikasi juga terintegrasi dengan model Convolutional Neural Network (CNN) untuk klasifikasi gambar, yang digunakan untuk merekomendasikan gaya rambut berdasarkan bentuk wajah dari gambar yang diunggah pengguna.
- Aplikasi menerima dan menyimpan file gambar yang diunggah pengguna ke direktori yang ditentukan (app.config['UPLOAD_PATH']).
- Implementasi session-based authentication untuk mengelola login pengguna. Endpoint login digunakan untuk memverifikasi dan mengotorisasi pengguna berdasarkan informasi login yang diberikan.
- Terdapat endpoint /chatbot yang menerima permintaan JSON dan memberikan jawaban berdasarkan model klasifikasi teks yang telah dilatih sebelumnya.
- Aplikasi mendukung manajemen pesanan pengguna dengan endpoint seperti /users_order dan /update_users_order, yang memungkinkan pengguna untuk membuat pesanan baru, memperbarui status pesanan, serta mengambil data pesanan berdasarkan ID.
- Pada mode debug (debug=True), aplikasi mencetak log aktivitas ke konsol, yang membantu dalam pemecahan masalah dan debugging selama pengembangan.
- Penggunaan Session: Menggunakan session untuk mengelola status login pengguna secara aman.

Flask

API ini untuk mendeteksi bentuk wajah dari gambar yang diunggah menggunakan model

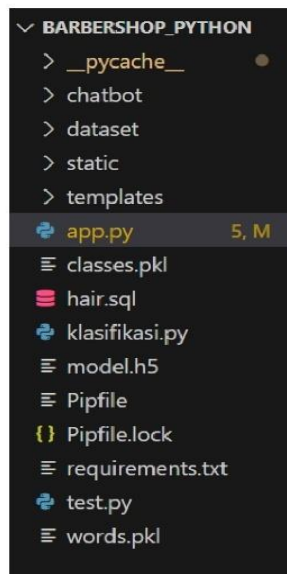
yang dilatih dengan TensorFlow. API menerima gambar, mendeteksi wajah, dan kemudian

mengklasifikasikan jenis kulit menggunakan model deep learning.

1. Model CNN dan Model Chatbot

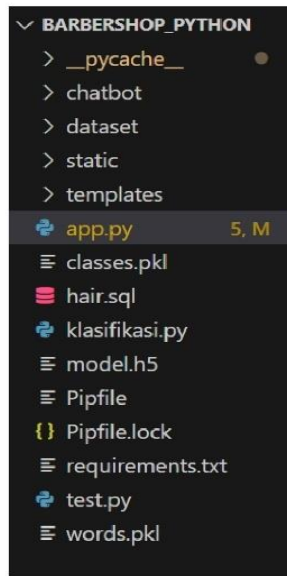
a. Model CNN

- Tempatkan file model (_model.h5) dalam folder `./barbershop_python/`



b. Model Chatbot

- Tempatkan file model (_words.pkl) dalam folder `./barbershop_python/`



2. App.py

- a. Memuat model CNN yang sudah di latih.

```
417  
418 if __name__ == "__main__":  
419     model.load_weights('./chatbot/chatbotmodel.h5')  
420     model_cnn.load_weights('./model.h5')
```

- b. Memuat model chatbot yang sudah di latih.

```
417  
418 if __name__ == "__main__":  
419     model.load_weights('./chatbot/chatbotmodel.h5')  
420     model_cnn.load_weights('./model.h5')
```

- c. Endpoint API untuk deteksi bentuk wajah.

Aplikasi Flask yang disediakan memiliki beberapa fitur utama dan komponen yang mendukung fungsionalitasnya. Berikut adalah deskripsi teknis untuk setiap komponen utama yang diimplementasikan dalam aplikasi:

i. Histori.dart

- AppBar: Menampilkan judul "History Pages" dengan warna teks amber dan latar belakang hitam. Ikon panah kembali (back) diubah warnanya menjadi putih.
- Container Utama: Menggunakan Container dengan latar belakang abu-abu dan padding 24. Di dalamnya terdapat ListView yang di-observe menggunakan Obx dari GetX untuk menampilkan data secara dinamis.
- Daftar Pesan: Menggunakan ListView untuk menampilkan daftar pesan yang diambil dari hisController.list. Setiap item ditampilkan menggunakan widget card.
- Widget Card: Menampilkan informasi dalam format card dengan latar belakang putih dan border hitam. Tanggal dan status ditampilkan dalam ListTile.

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:mobile/controller/helper.dart';
4 import 'package:mobile/controller/history.dart';
5
6 HistoryController hisController = Get.put(HistoryController());
7
8 class HistoryPages extends StatelessWidget {
9   const HistoryPages({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     hisController.unload();
14     return Scaffold(
15       backgroundColor: Colors.grey,
16       appBar: AppBar(
17         title: Text('History Pages', style: TextStyle(color: Colors.amber[600])),
18         centerTitle: true,
19         backgroundColor: Colors.black, // Change the background color to black
20
21         leading: IconButton(
22           icon: Icon(Icons.arrow_back,
23             color: Colors.white), // Change the back button color to white // Icon
24           onPressed: () => Navigator.of(context).pop(),
25         ), // IconButton // AppBar
26       body: Container(
27         color: Colors.grey,
28         padding: const EdgeInsets.all(24),
29         child: Obx(() => ListView(
30           shrinkWrap: true,
31           children: hisController.list.map((e) {
32             return Card(
33               shape: RoundedRectangleBorder(
34                 borderRadius: BorderRadius.circular(12.0)}, // Adjust radius as needed
35               child: ListTile(
36                 title: Text(e.tanggal),
37                 subtitle: Text('Status: ${e.status}'),
38               ), // ListTile
39             ), // Scaffold
40       ), // Scaffold
41     );
42
43   Widget card(String tanggal, String status) {
44     return Container(
45       margin: EdgeInsets.only(top: 8.0),
46       padding: EdgeInsets.all(12.0),
47       decoration: BoxDecoration(
48         color: Colors.white,
49         border: Border.all(
50           color: Colors.black,
51           width: 2.0), // Adjust color and width as needed // Border.all
52         borderRadius: BorderRadius.circular(12.0)}, // Adjust radius as needed
53       ), // BoxDecoration
54     child: ListTile(
55       title: Text(tanggal),
56       subtitle: Text('Status: $status'),
57     ), // ListTile
58   }
59 }
```

```

lib > widgets > chatbot.dart > _ChatbotState > getChatbotReply
1  import 'dart:convert';
2
3  import 'package:flutter/foundation.dart';
4  import 'package:flutter/material.dart';
5  import 'package:http/http.dart' as http;
6  import 'package:mobile/api/app.dart';
7
8  class Chatbot extends StatefulWidget {
9    _ChatbotState createState() => _ChatbotState();
10 }
11
12 class _ChatbotState extends State<Chatbot> {
13   final List<Message> _msg = <Messages>[];
14   final TextEditingController _txtController = TextEditingController();
15   void getChatbotReply(String userReply) async {
16     _txtController.clear();
17     var response = await http.post(Uri.parse(baseUrl.base + 'chatbot'),
18       headers: <String, String>{
19         'Content-Type': 'application/json; charset=UTF-8',
20       },
21       body: jsonEncode(<String, String>{
22         'message': userReply,
23       }));
24     var data = jsonDecode(response.body);
25     var botReply = data['answer'];
26     Message msg = Message(
27       text: botReply,
28       name: 'Bot',
29       type: false,
30     );
31     setState(() {
32       _msg.insert(0, msg);
33     });
34   }
35
36   void handleSubmitted(text) async {
37     if (kDebugMode) {
38       print(text);
39     }
40     _txtController.clear();
41     Message msg = Message(
42       text: text,
43       name: 'YOU',
44       type: true,
45     );
46     setState(() {
47       _msg.insert(0, msg);
48     });
49     getChatbotReply(text);
50   }
51 }
52
53 @override
54 Widget build(BuildContext context) {
55   return Scaffold(
56     backgroundColor: Colors.grey,
57     body: Container(
58       child: Column(
59         children: [
60           Container(
61             child: Wrap(
62               children: [
63                 AppBar(
64                   title: Text(
65                     'Chatbot',
66                   style: TextStyle(color: Colors.amber[600]),
67                 ), // Text
68                 centerTitle: true,
69                 backgroundColor: Colors.black,
70                 iconTheme: IconThemeData(color: Colors.white),
71               ), // AppBar
72             ), // Wrap
73           ), // Container
74           Flexible(
75             child: ListView.builder(
76               padding: const EdgeInsets.all(8.0),
77               reverse: true,
78               itemBuilder: (_, int index) => _msg[index],
79               itemCount: _msg.length,
80             ), // ListView.builder // Flexible
81             const Divider(height: 1.0),
82           ), Container(
83             decoration: BoxDecoration(color: Theme.of(context).cardColor),
84             child: IconTheme(
85               data: IconThemeData(color: Theme.of(context).cardColor),
86             ), child: Container(
87               margin: const EdgeInsets.symmetric(horizontal: 8.0),
88             ), child: Row(
89               children: [
90                 Flexible(
91                   child: TextField(
92                     controller: _txtController,
93                     onSubmitted: handleSubmitted,
94                     decoration: const InputDecoration.collapsed(
95                       hintText: 'Type message'), // InputDecoration.collapsed
96                   ), // TextField
97                 ), // Flexible
98               ), Container(
99                 margin: const EdgeInsets.symmetric(horizontal: 4.0),
100              ), child: IconButton(
101                icon: const Icon(Icons.send),
102                onPressed: () => handleSubmitted(_txtController.text),
103                color: Colors.amber,
104              ), // IconButton
105            ), // Row
106          ), // Container
107        ], // Column
108      ), // Container
109    ), // IconTheme
110  );

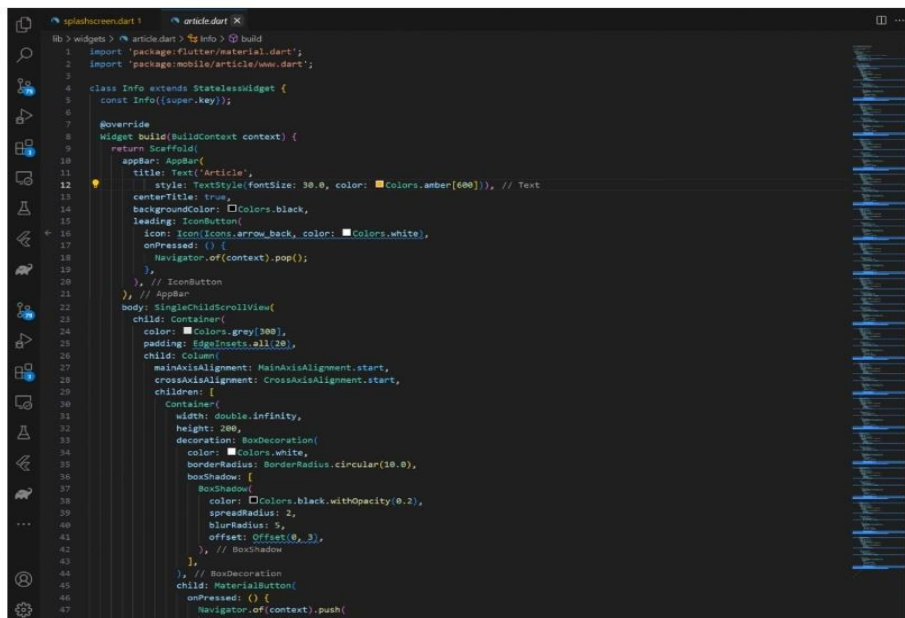
```

h. Chatbot.dart

- AppBar: Menampilkan judul "Chatbot" dengan gaya teks berwarna amber. AppBar ini terletak di bagian atas layar untuk memberikan navigasi dan informasi yang konsisten kepada pengguna.
- Container Utama: Memanfaatkan Column sebagai layout utama untuk menempatkan komponen-komponen aplikasi secara vertikal. Container ini menggunakan warna latar belakang abu-abu (grey) untuk estetika yang menarik.
- Daftar Pesan: Menggunakan ListView.builder untuk menampilkan daftar pesan yang diatur dalam List<Message>_msg. Setiap pesan ditampilkan dalam urutan terbalik sehingga pesan terbaru muncul di bagian atas layar.
- Input Pesan: Terdiri dari sebuah TextField yang digunakan pengguna untuk memasukkan pesan kepada bot. Saat pengguna menekan tombol "Send" atau mengirimkan pesan dengan cara lain, fungsi handleSubmitted dipanggil untuk mengirim dan menampilkan pesan tersebut.
- Balasan Bot: Setelah pengguna mengirim pesan, aplikasi menggunakan metode http.post untuk mengirimkan pesan ke server chatbot yang ditentukan. Hasil balasan dari server ditampilkan sebagai pesan dari bot dalam ListView.
- Widget Pesan (Message): Digunakan untuk menampilkan setiap pesan dalam aplikasi. Widget ini dapat menampilkan pesan yang dikirim oleh pengguna (type: true) dan balasan dari bot (type: false). Tampilan pesan pengguna menampilkan ikon avatar dengan latar belakang hitam dan ikon person_pin, sedangkan balasan bot menampilkan ikon avatar dengan latar belakang kuning dan ikon adb.

g. Article.dart

- AppBar: Memiliki judul "Article" dengan gaya teks berukuran 30 dan warna amber. Tombol kembali (back) berwarna putih untuk konsistensi tampilan.
- Container: Digunakan untuk menyimpan daftar artikel dengan layout Column, dengan SingleChildScrollView untuk memungkinkan pengguliran jika konten tidak muat pada layar.
- Tombol Artikel: Setiap artikel ditampilkan dalam sebuah Container yang memiliki bayangan (shadow), latar belakang putih, dan sudut yang dibulatkan. Setiap Container berisi sebuah MaterialButton dengan ikon yang sesuai dengan topik artikel dan teks judul artikel yang berukuran besar dan ditebalkan.
- Navigasi: Ketika pengguna menekan salah satu tombol artikel, aplikasi akan mengarahkan mereka ke halaman yang sesuai dengan artikel yang dipilih menggunakan MaterialPageRoute.

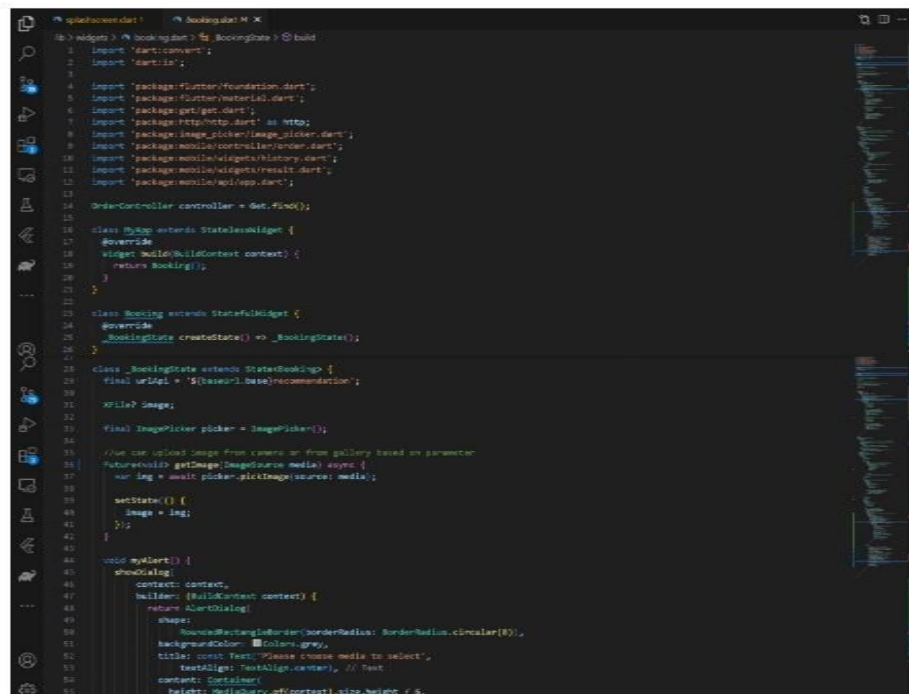


```
1 import 'package:flutter/material.dart';
2 import 'package:mobile/article/www.dart';
3
4 class Info extends StatelessWidget {
5   const Info({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10      appBar: AppBar(
11        title: Text('Article',
12          style: TextStyle(fontSize: 30.0, color: Colors.amber[600]), // Text
13          centerTitle: true,
14          backgroundColor: Colors.black,
15          leading: IconButton(
16            icon: Icon(Icons.arrow_back, color: Colors.white),
17            onPressed: () {
18              Navigator.of(context).pop();
19            }, // IconButton
20          ), // AppBar
21        ), // AppBar
22      body: SingleChildScrollView(
23        child: Container(
24          color: Colors.grey[300],
25          padding: EdgeInsets.all(20),
26          child: Column(
27            mainAxisAlignment: MainAxisAlignment.start,
28            crossAxisAlignment: CrossAxisAlignment.start,
29            children: [
30              Container(
31                width: double.infinity,
32                height: 200,
33                decoration: BoxDecoration(
34                  color: Colors.white,
35                  borderRadius: BorderRadius.circular(10.0),
36                  boxShadow: [
37                    BoxShadow(
38                      color: Colors.black.withOpacity(0.2),
39                      spreadRadius: 2,
40                      blurRadius: 5,
41                      offset: Offset(0, 3),
42                    ), // boxShadow
43                  ],
44                ), // BoxDecoration
45                child: MaterialButton(
46                  onPressed: () {
47                    Navigator.of(context).push(
48                      MaterialPageRoute
```


f. Booking.dart

Halaman ini menggunakan Flutter untuk memungkinkan pengguna mengunggah gambar dari galeri atau kamera, kemudian melakukan pengiriman gambar ke server untuk mendapatkan hasil rekomendasi.

- **AppBar:** Menggunakan AppBar dengan judul "Upload Image" yang diatur tengah, warna teks amber, dan latar belakang hitam. Tombol kembali (back) juga disesuaikan dengan warna putih.
- **Upload Gambar:** Pengguna dapat memilih untuk mengunggah gambar dari galeri atau kamera dengan menekan tombol yang sesuai. Gambar yang dipilih akan ditampilkan di layar.
- **Proses Upload:** Menggunakan `http.MultipartRequest` untuk mengirimkan gambar ke API yang disediakan (`urlApi`). Hasil yang diterima dari server berupa informasi tentang rekomendasi gaya rambut, yang ditampilkan dalam halaman `Result`.

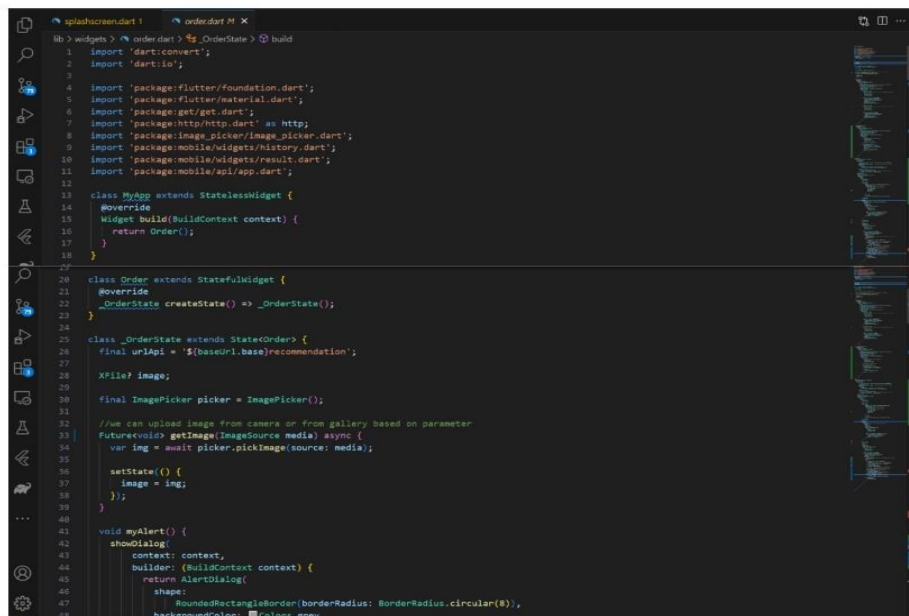


```
1 import 'dart:convert';
2 import 'dart:io';
3
4 import 'package:flutter/foundation.dart';
5 import 'package:flutter/material.dart';
6 import 'package:get/get.dart';
7 import 'package:http/http.dart' as http;
8 import 'package:image_picker/image_picker.dart';
9 import 'package:mobile/controller/order.dart';
10 import 'package:mobile/widgets/history.dart';
11 import 'package:mobile/widgets/result.dart';
12 import 'package:mobile/app.dart';
13
14 OrderController controller = Get.find();
15
16 class MyApp extends StatelessWidget {
17   @override
18   Widget build(BuildContext context) {
19     return Booking();
20   }
21 }
22
23 class Booking extends StatefulWidget {
24   @override
25   _BookingState createState() => _BookingState();
26 }
27
28 class _BookingState extends State<Booking> {
29   final urlApi = '${baseurl}base/recommendation';
30
31   Image? image;
32   ImagePicker picker = ImagePicker();
33
34   // You can upload image from camera or from gallery based on parameter
35   Future<void> getImage(ImageSource media) async {
36     var img = await picker.pickImage(source: media);
37
38     setState() {
39       image = img;
40     };
41   }
42
43   void showAlert() {
44     showDialog(
45       context: context,
46       builder: (BuildContext context) {
47         return AlertDialog(
48           shape:
49             RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
50           backgroundColor: Colors.grey,
51           title: const Text("Please choose media to select"),
52           text: const Text("Please choose media to select"),
53           content: Container(
54             height: MediaQuery.of(context).size.height / 4,
```

e. Order.dart

Halaman ini menggunakan Flutter untuk memungkinkan pengguna mengunggah gambar dari galeri atau kamera, kemudian melakukan pengiriman gambar ke server untuk mendapatkan hasil rekomendasi. AppBar: Menggunakan AppBar dengan judul "Upload Image" yang diatur tengah, warna teks amber, dan latar belakang hitam. Tombol kembali (back) juga disesuaikan dengan warna putih.

- Upload Gambar: Pengguna dapat memilih untuk mengunggah gambar dari galeri atau kamera dengan menekan tombol yang sesuai. Gambar yang dipilih akan ditampilkan di layar.
- Proses Upload: Menggunakan `http.MultipartRequest` untuk mengirimkan gambar ke API yang disediakan (`urlApi`). Hasil yang diterima dari server berupa informasi tentang jenis wajah dan rekomendasi gaya rambut, yang ditampilkan dalam halaman Result.



```
lib > widgets > order.dart | _OrderState > build
1 import 'dart:convert';
2 import 'dart:io';
3
4 import 'package:flutter/foundation.dart';
5 import 'package:flutter/material.dart';
6 import 'package:get/get.dart';
7 import 'package:http/http.dart' as http;
8 import 'package:image_picker/image_picker.dart';
9 import 'package:mobile/widgets/history.dart';
10 import 'package:mobile/widgets/result.dart';
11 import 'package:mobile/api/app.dart';
12
13 class MyApp extends StatelessWidget {
14   @override
15   Widget build(BuildContext context) {
16     return Order();
17   }
18 }
19
20
21 class Order extends StatefulWidget {
22   @override
23   _OrderState createState() => _OrderState();
24 }
25
26 class _OrderState extends State<Order> {
27   final urlApi = '${baseUrl.base}recommendation';
28
29   XFile? image;
30   final ImagePicker picker = ImagePicker();
31
32   //we can upload image from camera or from gallery based on parameter
33   Future<void> getImage(ImageSource media) async {
34     var img = await picker.pickImage(source: media);
35
36     setState(() {
37       image = img;
38     });
39   }
40
41   void myAlert() {
42     showDialog(
43       context: context,
44       builder: (BuildContext context) {
45         return AlertDialog(
46           shape:
47             RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
48           backgroundColor: Colors.grey,
```

d. Dashboard.dart

- App Bar: Menggunakan AppBar dengan judul "BarberBot" yang diatur tengah, warna teks amber, dan latar belakang hitam.
- Tombol Aksi: Menggunakan MaterialButton di dalam Container yang diberi dekorasi untuk menampilkan empat fungsi berbeda:
- Order: Mengarahkan pengguna ke halaman Order saat tombol "Order" diklik.
- Booking: Mengarahkan pengguna ke halaman Booking saat tombol "Booking" diklik.
- Article: Mengarahkan pengguna ke halaman Info (artikel) saat tombol "Article" diklik.
- Chatbot: Mengarahkan pengguna ke halaman Chatbot saat tombol "Chatbot" diklik.

```
lib > widgets > dashboard.dart > Dashboard > Dashboard
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:mobile/controller/helper.dart';
4 import 'package:mobile/widgets/booking.dart';
5 import 'package:mobile/widgets/chatbot.dart';
6 import 'package:mobile/widgets/article.dart';
7 import 'package:mobile/widgets/history.dart';
8 import 'package:mobile/widgets/login.dart';
9 import 'package:mobile/widgets/order.dart';
10
11 class Dashboard extends StatelessWidget {
12   const Dashboard({super.key});
13
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       title: 'BarberBot',
18       debugShowCheckedModeBanner: false,
19       home: Scaffold(
20         appBar: AppBar(
21           title: Text('BarberBot',
22             style: TextStyle(fontWeight: FontWeight.bold, color: Colors.amber[600]), // Text
23             centerTitle: true,
24             backgroundColor: Colors.black,
25             leading: IconButton(
26               onPressed: () => Get.to(const HistoryPages()),
27               icon: Icon(Icons.history, color: Colors.amber[600]), // IconButton
28             ),
29           actions: [
30             IconButton(
31               onPressed: () {
32                 HelperController.setShared('islogin', false, type: 'bool');
33                 Get.offAll(const LoginPage());
34               },
35               icon: Icon(
36                 Icons.logout,
37                 color: Colors.amber.shade600,
38               ) // Icon // IconButton
39             ),
40           ], // AppBar
41         ), // AppBar
42         body: Container(
43           width: 1000,
44           color: Colors.grey,
45           padding: const EdgeInsets.all(20),
46           child: Column(
47             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
48             children: [
49               Row(
50                 mainAxisAlignment: MainAxisAlignment.spaceEvenly,
51                 children: [
52                   Container(
53                     width: 150.0,
54                     height: 150.0,
55                     decoration: BoxDecoration(
56                       color: Colors.white,
57                       borderRadius: BorderRadius.circular(10.0),
58                       boxShadow: [
59
```

c. Splashscreen.dart

- `Get.put(OrderController());`: Menginisialisasi `OrderController` menggunakan `GetX` saat splash screen ditampilkan.
- Scaffold: Menampilkan splash screen dengan latar belakang berwarna abu-abu.
- Center: Menempatkan konten splash screen di tengah layar.
- Icon: Menampilkan ikon gunting dengan ukuran 100.0 dan warna amber.
- Text: Menampilkan nama aplikasi "BarberBot" dengan warna amber, ketebalan font tebal, dan ukuran font 30.0.
- Timer: Menunggu selama 3 detik sebelum mengalihkan pengguna ke halaman Dashboard menggunakan `Get.offAll`.

```
1 import 'dart:async';
2
3 import 'package:flutter/material.dart';
4 import 'package:get/get.dart';
5 import 'package:mobile/controller/dashboard.dart';
6 import 'package:mobile/controller/helper.dart';
7 import 'package:mobile/controller/order.dart';
8 import 'package:mobile/widgets/login.dart';
9
10 import 'dashboard.dart';
11
12
13 splashscreenStart() async {
14   DashboardController controller = Get.put(DashboardController());
15   Get.put(OrderController());
16
17   bool statusLogin =
18     await HelperController.getShared('isLoggedIn', type: 'bool') ?? false;
19   if (statusLogin) {
20     int id = await HelperController.getShared('id', type: 'int');
21     controller.id = id;
22     Get.offAll(() => const Dashboard());
23   } else {
24     Get.offAll(() => const LoginPage());
25   }
26 }
27
28 class SplashScreen extends StatelessWidget {
29   const SplashScreen({super.key});
30
31   @override
32   Widget build(BuildContext context) {
33     splashscreenStart();
34     return Scaffold(
35       backgroundColor: Colors.grey,
36       body: Center(
37         child: Column(
38           mainAxisAlignment: MainAxisAlignment.center,
39           crossAxisAlignment: CrossAxisAlignment.center,
40           children: <div data-bbox="501 922 541 939" data-label="Page-Footer">

E-17


```

b. Register.dart

- Controller untuk Input Pengguna: Membuat TextEditingController untuk menyimpan input nama, email, password, kota, kontak, alamat, dan gambar pengguna.
- Form Registrasi: Di Flutter, kita menggunakan widget TextField untuk mengumpulkan input pengguna dan widget ElevatedButton untuk mengirimkan data.
- Input untuk Nama, Email, Password, Kota, Kontak, Alamat, dan Gambar Pengguna: Menggunakan TextField dengan controller untuk setiap input yang diperlukan.
- Tombol Register: ElevatedButton yang ketika diklik akan mengirimkan form.

```
register.dart X
lib > widgets > register.dart > RegisterPage > @ build
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:mobile/controller/register.dart';
4 import 'package:mobile/widgets/login.dart';
5
6 RegisterController controller = Get.put(RegisterController());
7
8 class RegisterPage extends StatelessWidget {
9   @override
10  widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        backgroundColor: Colors.transparent,
14        elevation: 0,
15        leading: IconButton(
16          icon: const Icon(Icons.arrow_back, color: Colors.black),
17          onPressed: () {
18            Navigator.pop(context);
19          },
20        ), // IconButton
21        title: const Text(""),
22      ), // AppBar
23      body: Center(
24        child: Container(
25          width: double.infinity,
26          padding: const EdgeInsets.symmetric(horizontal: 24.0),
27          child: ListView(
28            children: [
29              const Text(
30                "Register",
31                style: TextStyle(
32                  fontSize: 24,
33                  fontWeight: FontWeight.bold,
34                  color: Colors.white,
35                ), // TextStyle
36            ), // Text
37            const SizedBox(height: 32),
38            TextFormField(
39              keyboardType: TextInputType.text,
40              decoration: InputDecoration(
41                labelText: "Username",
42                filled: true,
43                fillColor: Colors.orange[100],
44                border: OutlineInputBorder(
45                  borderRadius: BorderRadius.circular(8.0),
46                  borderSide: BorderSide.none,
47                ), // OutlineInputBorder
48              ), // InputDecoration
49              controller: controller.username,
50            ), // TextFormField
51            const SizedBox(height: 32),
52            TextFormField(
53              keyboardType: TextInputType.emailAddress,
54              decoration: InputDecoration(
```

The screenshot shows an IDE with the following components:

- File Explorer (Left):** Shows a project structure for a Flask application with folders like `static`, `templates`, and `models`. It also lists installed packages like `numpy`, `keras`, `tensorflow`, and `flask`.
- Terminal (Bottom):** Displays the command `python main.py` and the output of the application, including TensorFlow version information and package installation logs.
- Console (Right):** Shows the application's output, including a warning about development vs. production servers and the start of a TensorFlow session.

```

2024-07-18 08:07:57.575218: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly
different numerical results due to floating-point round-off errors from different computation orders. To turn them o
ff, set the environment variable 'TF_DISABLE_ONEINTE...'
2024-07-18 08:07:58.821859: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly
different numerical results due to floating-point round-off errors from different computation orders. To turn them o
ff, set the environment variable 'TF_DISABLE_ONEINTE...'
[nltk_data] Downloading package wordnet to
C:\Users\User\AppData\Local\Temp\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to
C:\Users\User\AppData\Local\Temp\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
C:\Users\User\AppData\Local\Temp\nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
C:\Users\User\AppData\Local\Programs\Python\Python312\lib\site-packages\keras\src\layers\convolutional\base_conv.py:1
47: UserWarning: Do not pass an 'input_shape' / 'input_dim' argument to a layer when using Sequential models, prefer u
sing an 'Input(shape)' object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
2024-07-18 08:08:02.121194: I tensorflow/core/platform/cpu_feature_guard.cc:218] This TensorFlow binary is optimized
to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler
flags.
C:\Users\User\AppData\Local\Programs\Python\Python312\lib\site-packages\keras\src\layers\conv\dense.py:87: UserWarning
: Do not pass an 'input_shape' / 'input_dim' argument to a layer when using Sequential models, prefer using an 'Input
(shape)' object as the first layer in the model instead.

```


Lampiran 6 Sertifikat HKI yang didapatkan


REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202471191, 25 Juli 2024

Pencipta

Nama : **Muhamad Ilham Nur Habibie Wibowo, Dega Surono Wibowo, S.T., M.Kom. dkk**

Alamat : **KP. Margaluyu RT.001.RW.014 Jelekong, Kecamatan Baleendah, Kab. Bandung, Propinsi Jawa Barat 40375, Baleendah, Bandung, Jawa Barat, 40375**

Kewarganegaraan : **Indonesia**

Pemegang Hak Cipta

Nama : **Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**

Alamat : **Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Tegal, Jawa Tengah 52142**

Kewarganegaraan : **Indonesia**

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Barberbot: Aplikasi Layanan Booking Barbershop Berbasis Android**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **25 Juli 2024, di Tegal**

Jangka waktu perlindungan : **Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.**

Nomor pencatatan : **000646542**

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri


IGNATIUS M.T. SILALAH
NIP. 196812301996031001



Disclaimer:
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhamad Ilham Nur Habibie Wibowo	KP. Margaluyu RT.001,RW.014 Jekekong, Kecamatan Baleendah, Kab. Bandung, Propinsi Jawa Barat 40375, Baleendah, Bandung
2	Dega Surono Wibowo, S.T., M.Kom.	Perum Sapphire Regency Blok H No.1, RT.004/RW.001, Kelurahan Pulosari, Kecamatan Brebes, Kabupaten Brebes, Propinsi Jawa Tengah 52213, Brebes, Brebes
3	Muhammad Fikri Hidayattullah. S.T., M.Kom.	Jl. Glatik No. 68, Randugunting, Kecamatan Tegal Selatan, Kota Tegal, Propinsi Jawa Tengah 52131, Tegal Selatan, Tegal



Lampiran 7 Lembar Bimbingan



SARJANA TERAPAN TEKNIK INFORMATIKA POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN SKRIPSI

Nama : Muahamad Ilham Nur Habibie Wibowo
 Nim : 20090069
 No. Ponsel : 088985044810
 Judul TA : Aplikasi Booking Barbershop : BarberBot Aplikasi Pemesanan Layanan Barbershop
 Dosen Pembimbing I : Dega Surono Wibowo, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1	25/3 2024	* Bilik UML sesuai standar ref. roni @rio arhan * Hkl.		f
2	4/4 2024	* UML fix * buat mock up/ desain aplikasi		f
3	24/4 2024	* Gilakan di develop		f
4	8/5 2024	* dilanjutka - sampai build app		f
5	25/5 2024	* lanjut ke order sambil booking + register + login		f
6	22/7 2024	coba coba bilik laporan kewangi + manual + teknis		f
7	1/8 2024	laporan cer silahkan perbaiki ke kewangi		f



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

D	2/8/2024	Laporan	Oh Silahkan salam	J

Tegal, 2-8-2024

Dosen Pembimbing I

Dega Suroho Wibowo, S.T., M.Kom.

NIPY. 06.014.183



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN SKRIPSI

Nama : Muahamad Ilham Nur Habibie Wibowo
 Nim : 20090069
 No. Ponsel : 088985044810
 Judul TA : Aplikasi Booking Barbershop : BarberBot Aplikasi Pemesanan Layanan Barbershop
 Dosen Pembimbing I : Dega Surono Wibowo, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	26/3 2024.	<ul style="list-style-type: none"> Bikin UML sesuai standar ref. rani @ rini arhan Hkl. 		f
*	4/4 2024.	<ul style="list-style-type: none"> UML. fix buat mockup/ desain aplikasi 		f-
*	24/4 2024.	<ul style="list-style-type: none"> Gilakan di develop 		f
*	8/5 2024	<ul style="list-style-type: none"> Silanjutnya - sampai build apk 		f
6	29/5 2024	<ul style="list-style-type: none"> lanjut ke order samban booking + registrasi logo 		f
	22/7 2024	<ul style="list-style-type: none"> coba coba bikin laporan ke range + manual + teknis 		f
7	1/8 2024	<ul style="list-style-type: none"> laporan ch silahkan perbaiki ke range 		f



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

6.	27/7	Prototipe	Ajudan HKI	
7	28/7	Laporan	Resnet ? Kuatun referensi	
8.	1/8	Laporan	Ace	