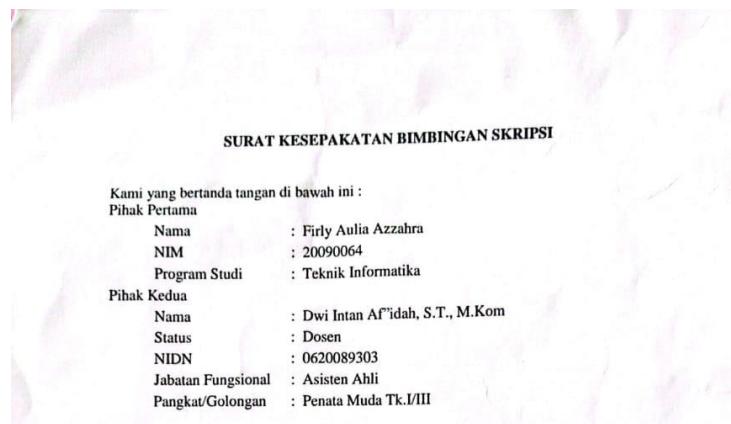


LAMPIRAN

Lampiran 1 Surat Kesediaan Pembimbing



Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Firly Aulia Azzahra
NIM : 20090064
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Dwi Intan Af'idah, S.T., M.Kom
Status : Dosen
NIDN : 0620089303
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : Penata Muda Tk.I/III

Pada hari ini Jumat tanggal 15 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I/II Skripsi Pihak Pertama dengan syarat adanya kemajuan pengerjaan skripsi yang dipresentasikan setiap dua minggu. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 15 Maret 2024

Pihak Pertama

Firly Aulia Azzahra

Pihak Kedua

Dwi Intan Af'idah, S.T., M.Kom

Mengetahui
Ketua Program Studi Sajidna Umaran 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 : Firly Aulia Azzahra
NIM : 20090064
Program Studi : Teknik Informatika

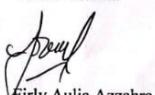
Pihak Kedua

Nama : Ardi Susanto, S.Kom., M.Cs
Status : Dosen
NIDN : 0629109301
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : Penata Muda Tk.I/III/b

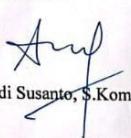
Pada hari ini Jumat tanggal 15 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I/II Skripsi Pihak Pertama dengan **syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua**. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 18 Maret 2024

Pihak Pertama


Firly Aulia Azzahra

Pihak Kedua


Ardi Susanto, S.Kom., M.Cs

Mengetahui
Ketua Program Studi Sistem Informasi Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom
NIPY .09.015.225

Lampiran 2. Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

- 1 Nama : Firly Aulia Azzahra
Kewarganegaraan : Indonesia
Alamat : Jalan Cempaka No. 48, RT/RW 10/01, Mangkukusuman, Tegal
Timur, Provinsi Jawa Tengah
- 2 Nama : Ardi Susanto, S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Margadana, Tegal
- 3 Nama : Dwi Intan Af'ida, M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Grinting RT/RW 003/001, Kecamatan Bulakamba,
Kabupaten Brebes, Provinsi Jawa Tengah

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:

Berupa : Program Komputer
Berjudul : Aplikasi Korelasi Warna Pakaian Berdasarkan Warna Kulit Menggunakan
Metode K-Means Clustering dan YOLOV8 Berbasis Mobile

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:
 - a. Permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. 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.
 - c. 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, Juli 2024

firly Aulia Azzahra)

(Ardi Susanto, S.Kom., M.Cs.)

(Dwi Intan Afidah, M.Kom)

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1	Nama : Firly Aulia Azzahra
	Kewarganegaraan : Indonesia
	Alamat : Jalan Cempaka No.48 RT/RW 10/01 Mangkukusuman, Tegal Timur, Provinsi Jawa Tengah
2	Nama : Ardi Susanto, S.Kom., M.Cs.
	Kewarganegaraan : Indonesia
	Alamat : Margadana, Tegal
3	Nama : Dwi Intan Af'ida, M.Kom
	Kewarganegaraan : Indonesia
	Alamat : Desa Grinting RT/RW 003/001, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah

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

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

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Korelasi Warna Pakaian Berdasarkan Warna Kulit Menggunakan Metode K-Means Clustering dan YOLOv8 Berbasis Mobile", 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, Juli 2024
Pencipta

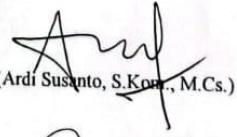
Pemegang Hak Cipta
Ketua P3M



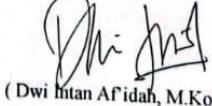
(Dr. Addi Budi Riyanta, S.Si., M.T)



(Firly Aulia Azzahra)



(Ardi Susanto, S.Kom., M.Cs.)



(Dwi Intan Af'ida, M.Kom)

Lampiran 4 Syarat Pengajuan HKI (Mannual Book & Technical Book)



DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
1. PENDAHULUAN.....	4
1.1. Tujuan Pembuatan Dokumen.....	4
1.2. Deskripsi Umum Aplikasi.....	4
1.3. Deskripsi Dokumen	4
2. PERANGKAT YANG DIBUTUHKAN.....	4
2.1. Perangkat Lunak	4
2.2. Perangkat Keras	5
3. MENU DAN CARA PENGGUNAAN.....	5
3.1. Struktur Menu	5
3.2. Cara Penggunaan.	6
1. <i>Splash Screen</i>	6
2. <i>Login</i>	7
3. Daftar Akun.....	8
4. Lupa Kata Sandi.....	10
5. Halaman Deteksi.....	11
6. Halaman Outfit.....	12
7. Halaman Detail Outfit.....	13
8. Halaman Detail Warna	14

DAFTAR GAMBAR

Gambar 1. Splash Screen	7
Gambar 2. Tampilan Login	8
Gambar 3. Tampilan Halaman Daftar Akun	9
Gambar 4. Tampilan Lupa Kata Sandi.....	11
Gambar 5. Tampilan Halaman Deteksi	12
Gambar 6. Tampilan Halaman Outfit.....	13
Gambar 7. Halaman Detail Outfit.....	14
Gambar 8. Halaman Detail Warna	16

1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Dokumen *user manual* ini dibuat untuk memberikan penjelasan dan gambaran atau sebagai panduan kepada pengguna mengenai penggunaan Aplikasi Mobile F-SkinHue.

1.2. Deskripsi Umum Aplikasi

Aplikasi ini dirancang untuk memberikan rekomendasi perpaduan warna outfit berdasarkan warna kulit, warna kulit diklasifikasikan menjadi tiga yaitu Cool, Neutral, dan Warm. Terdapat juga fitur klasifikasi warna, warna yang diklasifikasikan yaitu Merah, Hijau, dan Biru. Saat warna sudah diklasifikasikan maka akan menampilkan rekomendasi kombinasi warna yang sesuai. Klasifikasi pada aplikasi F-SkinHue menggunakan YOLOv8.

1.3. Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan penggunaan Aplikasi Mobile F-SkinHue.

1. Bagian 1

Bagian ini berisi informasi umum yang meliputi tujuan pembuatan dokumen, deskripsi umum sistem, dan deskripsi dokumen.

2. Bagian 2

Bagian ini berisi perangkat yang dibutuhkan untuk penggunaan aplikasi meliputi perangkat keras dan perangkat lunak.

3. Bagian 3

Bagian ini berisi panduan atau cara penggunaan.

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak

Untuk menjalankan aplikasi ini, berikut beberapa perangkat lunak yang dibutuhkan yaitu:

1. Sistem Operasi Android. Aplikasi ini harus dijalankan pada perangkat yang menggunakan sistem operasi Android.

2. Koneksi Internet. Untuk menjalankan aplikasi dengan lancar, diperlukan koneksi internet yang stabil.
3. Penyimpanan Data.

2.2. Perangkat Keras

Selain perangkat lunak, aplikasi ini juga memerlukan perangkat keras, terutama smartphone Android. Smartphone Android digunakan untuk menjalankan aplikasi F-SkinHue.

3. MENU DAN CARA PENGGUNAAN

3.1. Struktur Menu

Berikut struktur menu yang ada pada aplikasi SideFish adalah sebagai berikut:

1. Splash Screen

Tampilan awal saat aplikasi dibuka. Menampilkan logo aplikasi sebelum pengguna diarahkan ke halaman *login*.

2. Login

Menu ini digunakan untuk masuk ke aplikasi dengan akun yang sudah terdaftar. Pengguna harus memasukkan email dan kata sandi yang telah didaftarkan.

3. Daftar Akun

Menu ini digunakan untuk mendaftarkan akun baru. Pengguna harus memasukkan nama, email, dan kata sandi untuk mendaftar akun baru.

4. Login

Menu ini digunakan untuk masuk ke aplikasi dengan akun yang sudah terdaftar. Pengguna harus memasukkan email dan kata sandi yang telah didaftarkan.

5. Register

Menu ini digunakan untuk mendaftarkan akun baru. Pengguna harus memasukkan nama, email, dan kata sandi untuk mendaftar akun baru.

6. Lupa Kata Sandi

Menu ini untuk mereset kata sandi jika pengguna lupa dengan kata sandinya.

7. Deteksi

Halaman ini untuk mengunggah gambar atau mengambil gambar kulit maupun warna untuk dideteksi, jika deteksi berhasil maka diarahkan ke halaman rekomendasi warna outfit atau rekomendasi perpaduan warna.

8. Outfit

Tampilan ini muncul saat pengguna memilih deteksi skin tone pada halaman deteksi, lalu muncul klasifikasi skin tonenya dan muncul beragam outfit yang cocok untuk skin tone yang terdeteksi

9. Detail Outfit

Menu ini digunakan untuk melihat outfit yang ditampilkan berdasarkan hasil deteksi. Jika pengguna memilih outfit, mereka akan diarahkan ke halaman ini yang berisi foto outfit, detail kombinasi outfit, dan tiga palet kombinasi warna.

10. Detail Warna

Menu ini digunakan untuk menampilkan kombinasi warna dari hasil deteksi warna, halaman ini menampilkan rekomendasi kombinasi warna berdasarkan klasifikasi warna yang terdeteksi, terdapat foto palet warna beserta deskripsi kombinasi warnanya.

3.2. Cara Penggunaan.

1. *Splash Screen*

Buka aplikasi dan pastikan *smartphone* terhubung ke internet. Tunggu beberapa saat. Tampilan tersebut dapat dilihat pada Gambar 1.



Gambar 1. *Splash Screen*

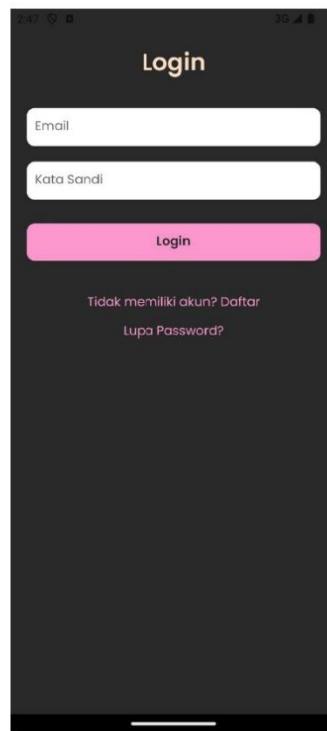
2. *Login*

Pada halaman ini, Anda dapat masuk ke aplikasi dengan menggunakan email dan kata sandi yang telah didaftarkan sebelumnya. Tampilan tersebut dapat dilihat pada Gambar 2. Berikut langkah-langkah nya:

- 1) Masukkan Email dan Kata Sandi. Pastikan sudah terdaftar dan terverifikasi.
- 2) Klik button Masuk.
- 3) Aplikasi akan menampilkan halaman Deteksi

Catatan:

Jika belum melakukan pendaftaran, Anda bisa klik text “Tidak memiliki akun? Daftar” dan akan menampilkan halaman pendaftaran akun.

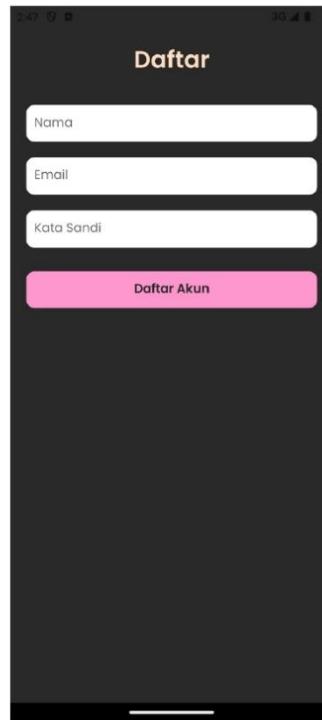


Gambar 2. Tampilan *Login*

3. Daftar Akun

Pada halaman ini, Anda akan diarahkan untuk membuat akun baru supaya dapat mengakses semua fitur pada aplikasi. Tampilan tersebut dapat dilihat pada Gambar 3. Berikut langkah-langkah nya:

- 1) Terdapat empat formular untuk proses pendaftaran, masukkan informasi yang diperlukan seperti Nama, Email, dan Kata Sandi.
- 2) Klik button “Buat Akun”
- 3) Jika pendaftaran berhasil akan dialihkan ke halaman login.

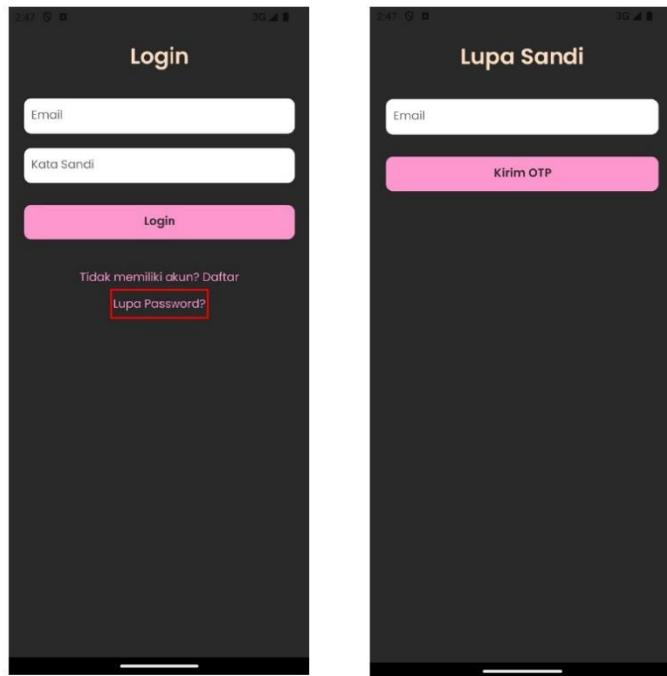


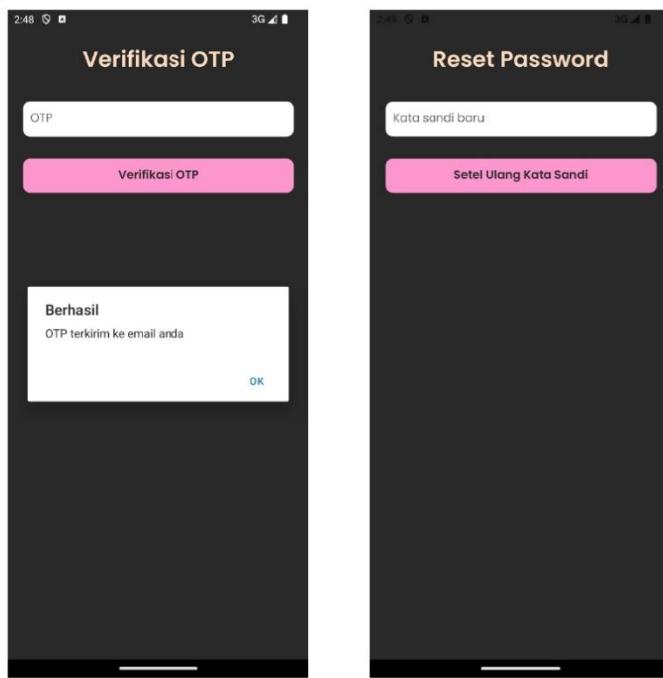
Gambar 3. Tampilan Halaman Daftar Akun

4. Lupa Kata Sandi

Pada halaman lupa kata sandi merupakan halaman untuk mereset kata sandi jika pengguna lupa dengan kata sandinya. Tampilan tersebut dapat dilihat pada Gambar 4. Berikut langkah-langkah yang harus dilakukan:

- 1) Masuk ke halaman login.
- 2) Tekan text “Lupa Password”.
- 3) Lalu masukkan email yang pernah terdaftar, dan akan mendapatkan kode OTP melalui email.
- 4) Masukkan kode OTP.
- 5) Jika verifikasi otp berhasil maka akan muncul reset kata sandi.

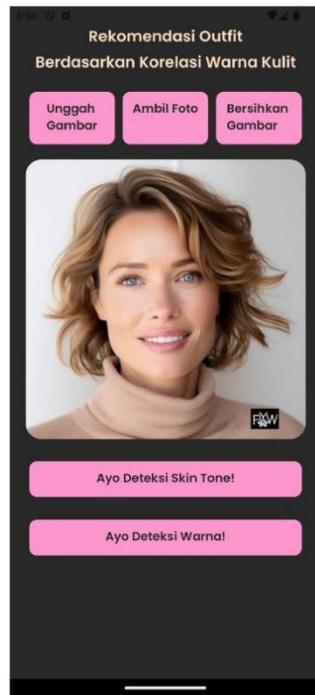




Gambar 4. Tampilan Lupa Kata Sandi

5. Halaman Deteksi Skin Tone

- 1) Masuk ke halaman Setelah login, aplikasi akan menampilkan halaman Deteksi untuk klasifikasi gambar dengan cara mengunggah foto atau mengambil foto secara langsung. Tampilan tersebut dapat dilihat pada Gambar 5.
- 2) Pada halaman ini terdapat unggah gambar, ambil foto dan bersihkan gambar
- 3) Tekan tombol unggah gambar atau ambil foto
- 4) Lalu masukkan gambar atau ambil foto
- 5) Klik tombol Deteksi Skin Tone



Gambar 5. Tampilan Halaman Deteksi

6. Halaman Outfit

Pada halaman ini, setelah pengguna mengunggah atau mengambil gambar dan menekan tombol "Deteksi Skin Tone", mereka akan diarahkan ke halaman Outfit. Outfit yang ditampilkan berdasarkan hasil deteksi, yaitu Cool, Neutral, atau Warm. Tampilan deteksi skin tone.

tersebut dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Halaman Outfit

7. Halaman Detail Outfit

Pada halaman ini terdapat detail rekomendasi outfit yang dipilih pengguna dari halaman outfit. Halaman Detail Outfit berisi foto outfit, detail kombinasi outfit, dan tiga palet kombinasi warna. Berikut tampilan pada halaman ini dapat dilihat pada Gambar 7.



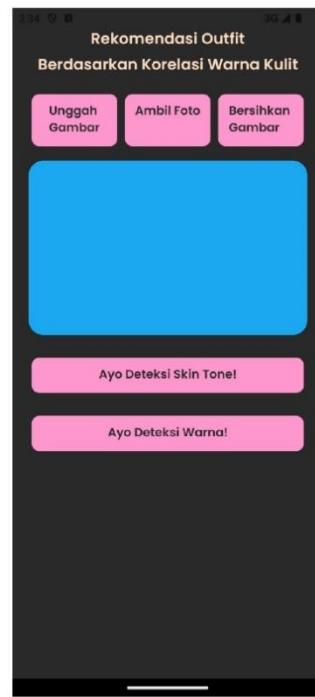
Gambar 7. Halaman Detail Outfit

8. Deteksi Warna

Pada halaman Deteksi untuk klasifikasi gambar dengan cara mengunggah foto atau mengambil foto secara langsung. Tampilan tersebut dapat dilihat pada Gambar 8.

- 1) Masuk ke halaman deteksi warna.

- 2) Pada halaman ini terdapat unggah gambar, ambil foto dan bersihkan gambar
- 3) Tekan tombol unggah gambar atau ambil foto
- 4) Lalu masukkan gambar atau ambil foto
- 5) Klik tombol “Ayo Deteksi Warna”



Gambar 8. Halaman Deteksi Warna

9. Halaman Deteksi Warna

Menu ini digunakan untuk menampilkan kombinasi warna dari hasil deteksi warna, halaman ini menampilkan rekomendasi kombinasi warna berdasarkan klasifikasi warna yang terdeteksi, terdapat foto palet warna beserta deskripsi

kombinasi warnanya. Berikut tampilan halaman menu makan harian dapat dilihat pada Gambar 9.



Gambar 9. Halaman Detail Warna

DOKUMEN TEKNIS

F-SKINHUE

APLIKASI KORELASI WARNA PAKAIAN
BERDASARKAN WARNA KULIT
MENGGUNAKAN METODE K-MEANS
CLUSTERING DAN YOLOV8 BERBASIS
MOBILE



Oleh :

Firly Aulia Azzahra

Ardi Susanto, S.Kom, M.Cs.

Dwi Intan Af'idah, S.T., M.Kom.



1 Profil

F-SkinHue adalah Aplikasi deteksi korelasi warna berdasarkan warna kulit berbasis mobile menggunakan metode YOLOv8 dengan solusi inovatif yang dirancang untuk mempermudah pengguna dalam menemukan undertone kulit dan rekomendasi warna berdasarkan undertone. Dengan memanfaatkan teknologi deep learning dan pengolahan citra, aplikasi ini mampu mengenali berbagai jenis undertone secara akurat. Pengguna cukup mengambil foto atau mengunggah gambar wajah, dan aplikasi akan memberikan informasi lengkap tentang rekomendasi warna outfit berdasarkan undertone mereka. Terdapat juga fitur tambahan yaitu deteksi warna dengan output rekomendasi palet warna yang cocok dengan warna yang dideteksi.

2 Latar Belakang

Dalam dunia *fashion* dan kecantikan, memilih warna yang tepat berdasarkan undertone kulit adalah faktor penting untuk meningkatkan penampilan dan rasa percaya diri. Namun, banyak orang merasa kesulitan dalam menentukan undertone kulit mereka dan memilih warna yang sesuai. Masalah ini memerlukan solusi yang inovatif dan mudah diakses untuk membantu individu dalam membuat keputusan yang lebih baik mengenai pilihan warna.

F-SkinHue hadir sebagai jawaban atas kebutuhan tersebut. Aplikasi berbasis mobile ini menggunakan metode YOLOv8 yang canggih untuk mendekripsi korelasi warna berdasarkan warna kulit. Dengan memanfaatkan teknologi deep learning dan pengolahan citra, F-SkinHue mampu mengenali berbagai jenis undertone secara akurat. Inovasi ini memungkinkan pengguna untuk menemukan undertone kulit mereka hanya dengan mengambil foto atau mengunggah gambar wajah mereka ke aplikasi.

Selain deteksi undertone kulit, F-SkinHue juga memberikan rekomendasi warna outfit yang disesuaikan dengan undertone pengguna. Fitur tambahan lainnya adalah deteksi warna dari gambar dengan *output* rekomendasi palet warna yang cocok. Ini

memberikan nilai tambah bagi pengguna yang ingin mengeksplorasi palet warna yang sesuai dengan preferensi dan kebutuhan mereka.

Pengembangan F-SkinHue didasarkan pada pemahaman bahwa setiap individu memiliki kebutuhan unik dalam memilih warna yang tepat untuk meningkatkan penampilan mereka. Dengan memberikan solusi yang mudah digunakan dan akurat, F-SkinHue bertujuan untuk membantu pengguna membuat keputusan yang lebih baik dalam hal *fashion* dan kecantikan.

Dengan demikian, F-SkinHue bukan hanya sekadar aplikasi, tetapi juga sebuah alat yang dapat membantu individu merasa lebih percaya diri dan nyaman dengan pilihan warna mereka, menggabungkan teknologi canggih dengan kebutuhan sehari-hari dalam *fashion* dan kecantikan.

3 Manfaat

Berikut manfaat dari aplikasi ini:

- a. Deteksi Undertone yang Akurat: Memanfaatkan teknologi deep learning dan metode YOLOv8, aplikasi ini mampu mendeteksi undertone kulit dengan tingkat akurasi tinggi, membantu pengguna memahami jenis undertone mereka secara tepat.
- b. Rekomendasi Warna yang Tepat: F-SkinHue memberikan rekomendasi warna outfit berdasarkan hasil deteksi undertone, memudahkan pengguna dalam memilih warna pakaian yang paling sesuai dengan kulit mereka untuk meningkatkan penampilan.
- c. Peningkatan Rasa Percaya Diri: Dengan mendapatkan informasi yang tepat mengenai warna-warna yang cocok dengan undertone kulit mereka, pengguna dapat merasa lebih percaya diri dalam berpakaian dan tampil di berbagai kesempatan.
- d. Fitur Deteksi Warna Tambahan: Selain rekomendasi warna outfit, aplikasi ini juga dilengkapi dengan fitur deteksi warna dari gambar, memberikan

- rekомендација палета боја која је уједињена са бојом која је детектована, што ће корисника омогућити да истражи већи број изабраних боја.
- e. Удобност коришћења: Апликација је дизајнирана да користи једноставан и кориснички пријатељски интерфејс, што ће кориснику омогућити лако да сними фотографију или пријави слику њених лица да би добијао резултате детекције и рекомендације боја.
 - f. Практична и иновативна решења: F-SkinHue омогућава кориснику практичне решење која су обично узроковане смештавањем боја која не сагодију њену кожу. Ово је посебно корисно за људе са људском кожом, који често се срећу са тешкоћама при одабиру боја.
 - g. Подршка моднијим и привлачним изгледима: F-SkinHue помаже кориснику да нађе боје које сагодију њену кожу и које ће да подсећају на њену кожу. Ово је посебно корисно за кориснике који користе F-SkinHue за обично коришћење, као и за посебне прилике као што су свечане прилике.

4 Spesifikasi Teknis

Berikut uraian spesifikasi untuk pembuatan aplikasi F-SkinHue:

- a. Visual Studio Code
- b. Anaconda
- c. Google Colaboratory
- d. Android Studio
- e. React Native

5 Source Code

1) Logic

- 1. Помоделан

- a. Метод K-means Clustering

Помоделан са K-means clustering-ом за класификацију skin Tone апликације F-SkinHue добијајући резултати са високом акуратношћу као што су приказано испод.

K-means model saved to /content/drive/MyDrive/SKRIPSI FIRLY/model/kmeans_model.joblib					
Label encoder saved to /content/drive/MyDrive/SKRIPSI FIRLY/model/label_encoder.joblib					
Validation Classification Report:					
	precision	recall	f1-score	support	
Cool	0.05	0.04	0.05	25	
Neutral	0.35	0.25	0.29	36	
Tidak Diketahui	0.10	0.13	0.11	31	
Warm	0.21	0.27	0.24	30	
accuracy			0.18	122	
macro avg	0.18	0.17	0.17	122	
weighted avg	0.19	0.18	0.18	122	

b. Metode YOLOv8 (Skin Tone)

Pemodelan dengan YOLOv8 untuk klasifikasi skin Tone aplikasi F-SkinHue mendapatkan hasil akurasi seperti gambar berikut.

```

classes      top1_acc      top5_acc      100% 7/7 [00:00:00.000, 7.40it/s]
all          0.85          1

Epoch      GPU mem      loss      Instances      Size
3/4        1.02G      0.4988      4      224: 100% 77/77 [00:14<00:00, 5.22it/s]
classes    top1_acc      top5_acc      100% 7/7 [00:00:00.00, 9.08it/s]
all          0.875         1

Epoch      GPU mem      loss      Instances      Size
4/4        1.06G      0.4732      4      224: 100% 77/77 [00:15<00:00, 4.94it/s]
classes    top1_acc      top5_acc      100% 7/7 [00:00:00.00, 7.25it/s]
all          0.895         1

4 epochs completed in 0.028 hours.
Optimizer stripped from runs/classify/train/weights/last.pt, 31.7M
Optimizer stripped from runs/classify/train/weights/best.pt, 31.7M

Validating runs/classify/train/weights/best.pt...
Ultralytics YOLOv8.2.49 ✘ Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MB)
YOLOv8m-cls summary (fused): 103 layers, 15767780 parameters, 0 gradients, 41.6 GFLOPs
train: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/train... found 1220 images in 4 classes ✘
val: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/val... found 200 images in 4 classes ✘
test: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/test... found 40 images in 4 classes ✘
classes    top1_acc      top5_acc      100% 7/7 [00:01:00:00, 3.97it/s]
all          0.895         1
Speed: 0.2ms preprocess, 1.0ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs/classify/train
Results saved to runs/classify/train
💡 Learn more at https://docs.ultralytics.com/modes/train

```

c. Metode YOLOv8 (Warna)

Pemodelan dengan YOLOv8 untuk klasifikasi warna aplikasi

F-SkinHue mendapatkan hasil akurasi seperti gambar berikut.

```

Ultralytics YOLOv8.2.49 ✘ Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MB)
YOLOv8m-cls summary (fused): 103 layers, 15767780 parameters, 0 gradients, 41.6 GFLOPs
train: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/train... found 1220 images in 4 classes ✘
val: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/val... found 200 images in 4 classes ✘
test: /content/drive/MyDrive/SKRIPSI FIRLY/YOLOv8/dataset/test... found 40 images in 4 classes ✘
classes    top1_acc      top5_acc      100% 7/7 [00:01:00:00, 3.97it/s]
all          0.895         1
Speed: 0.2ms preprocess, 1.0ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs/classify/val
Results saved to runs/classify/train
1.0

```

2. Route Endpoint API

Bagian kode ini mendefinisikan konfigurasi rute API menggunakan Flask untuk aplikasi F-SkinHue. Di dalam aplikasi Flask, terdapat rute yang digunakan untuk memproses model YOLOv8 dalam klasifikasi gambar, serta data JSON untuk rekomendasi outfit.

```
# YOLO v8
model_path = '/Users/filry/flask/model/best.pt'
model = YOLO(model_path)

# YOLO v8 warna
model_path_warna = '/Users/filry/flask/model/warna.pt'
model_warna = YOLO(model_path_warna)

def classify_image(image_path):
    results = model(image_path)
    names_dict = results[0].names
    probs = results[0].probs.data.tolist()
    predicted_class = names_dict[np.argmax(probs)]
    return predicted_class

def classify_warna_image(image_path):
    results = model_warna(image_path)
    names_dict = results[0].names
    probs = results[0].probs.data.tolist()
    predicted_class = names_dict[np.argmax(probs)]
    return predicted_class
```

```

@app.route('/upload', methods=['POST'])
def upload_files():
    if 'files' not in request.files:
        return jsonify({'error': 'No files uploaded'}), 400
    files = request.files.getlist('files')
    if not files:
        return jsonify({'error': 'No files uploaded'}), 400
    results = []
    for file in files:
        if file.filename == '':
            return jsonify({'error': 'No file selected'}), 400
        filepath = os.path.join(UPLOAD_FOLDER, secure_filename(file.filename))
        file.save(filepath)
        predicted_class = classify_image(filepath)
        result = {
            'class': predicted_class
        }
        results.append(result)
    return jsonify(results)

@app.route('/classname', methods=['POST'])
def upload_name_file():
    if 'files' not in request.files:
        return jsonify({'error': 'No files uploaded'}), 400
    files = request.files.getlist('files')
    if not files:
        return jsonify({'error': 'No file uploaded'}), 400
    results = []
    for file in files:
        if file.filename == '':
            return jsonify({'error': 'No file selected'}), 400
        filepath = os.path.join(UPLOAD_FOLDER, secure_filename(file.filename))
        file.save(filepath)
        predicted_class = classify_name_image(filepath)  -> You, 33 minutes ago -
        result = {
            'class': predicted_class
        }
        results.append(result)
    return jsonify(results)

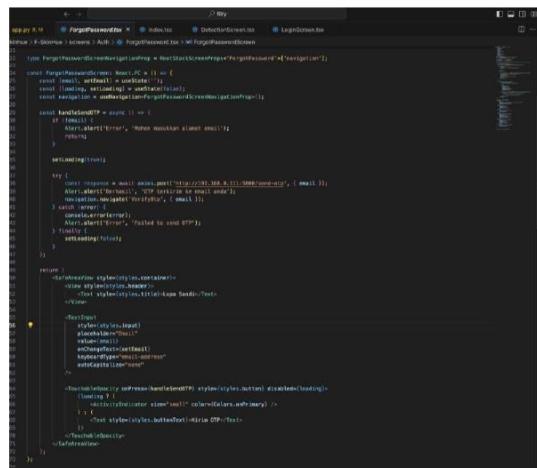
categories_blueprint.route('/api/json/categories/<categories>', methods=['GET'])
def get_categories(categories):
    if categories == 'Cool':
        outfit = [
            {
                "idOutfit": "1801",
                "strOutfit": "Outfit Cool 1",      -> famil ahead, last week - first
                "strOutfitThumb": "http://192.168.1.7:5000/img/C1.jpg",
            },
            {
                "idOutfit": "1802",
                "strOutfit": "Outfit Cool 2",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C2.jpg",
            },
            {
                "idOutfit": "1803",
                "strOutfit": "Outfit Cool 3",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C3.jpg",
            },
            {
                "idOutfit": "1804",
                "strOutfit": "Outfit Cool 4",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C4.jpg",
            },
            {
                "idOutfit": "1805",
                "strOutfit": "Outfit Cool 5",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C5.jpg",
            },
            {
                "idOutfit": "1806",
                "strOutfit": "Outfit cool 6",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C6.jpg",
            },
            {
                "idOutfit": "1807",
                "strOutfit": "Outfit Cool 7",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C7.jpg",
            },
            {
                "idOutfit": "1808",
                "strOutfit": "Outfit Cool 8",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C8.jpg",
            },
            {
                "idOutfit": "1809",
                "strOutfit": "Outfit Cool 9",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C9.jpg",
            },
            {
                "idOutfit": "1800",
                "strOutfit": "Outfit Cool 10",
                "strOutfitThumb": "http://192.168.1.7:5000/img/C10.jpg",
            }
        ]
        return jsonify(outfit)
    else:
        return jsonify({'error': 'Category not found'}), 404

```

2) User Interface

1. Login

Halaman login ini menyajikan sebuah formulir dengan dua input utama, yaitu email dan password. Masing-masing dikelola oleh ‘TextInput’ untuk mengontrol teks yang diinput pengguna, serta divalidasi untuk memastikan data diisi dengan benar. Halaman juga menyediakan tombol Masuk yang menggunakan fungsi ‘handleLogin’ untuk mengirim data ke server melalui API.



```
import React, { useState, useEffect } from 'react';
import { Form, Input, Button } from 'antd';
import axios from 'axios';

const LoginPage = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);

  const handleLogin = async () => {
    if (email === '' || password === '') {
      setError('Please fill in email and password');
      return;
    }

    setLoading(true);

    try {
      const response = await axios.post('http://127.0.0.1:5000/login', { email });
      if (response.data.message === 'Email Verified') {
        setError(null);
        setLoading(false);
      }
    } catch (error) {
      setError('Failed to send OTP');
    }
  };

  return (
    <Form
      onFinish={handleLogin}
      style={{ width: '100%', margin: 'auto' }}
    >
      <Form.Item
        style={{ width: '100%' }}
        label="Email"
        name="email"
        rules={[{ required: true, message: 'Email is required' }]}
      >
        <Input
          style={{ width: '100%' }}
          placeholder="Email"
          onChange={e => setEmail(e.target.value)}
        />
      </Form.Item>
      <Form.Item
        style={{ width: '100%' }}
        label="Password"
        name="password"
        rules={[{ required: true, message: 'Password is required' }]}
      >
        <Input
          style={{ width: '100%' }}
          type="password"
          placeholder="Password"
          onChange={e => setPassword(e.target.value)}
        />
      </Form.Item>
      <Form.Item
        style={{ width: '100%' }}
        label="Login"
        name="submit"
        rules={[{ required: true, message: 'Login is required' }]}
        onFinish={handleLogin}
      >
        <Button type="primary" style={{ width: '100%' }}>Login</Button>
      </Form.Item>
    </Form>
  );
}

export default LoginPage;
```

2. Daftar Akun

Pada halaman Daftar Akun terdapat tiga field untuk mengisi form pendaftaran, yaitu Nama, Email, dan Kata Sandi. Pada halaman ini menggunakan endpoint/register untuk method POST agar terkirim ke backend flask.

```

File / F-Storage / screens / Auth / 01 RegisterScreens / 01 RegisterScreen / 01 handleRegister
const RegisterScreens = () => {
  const [username, setUsername] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [loading, setLoading] = useState(false);
  const navigation = useNavigation();

  const handleRegister = async () => {
    if (!username || !email || !password) {
      alert('Error', 'Please complete all fields');
      return;
    }

    setLoading(true);
    try {
      const response = await axios.post('http://192.168.1.7:5000/register', {
        username,
        email,
        password
      });

      Alert.alert('Success', 'User registered successfully');
      navigation.navigate('Login');
    } catch (error) {
      console.error(error);
      Alert.alert('Error', 'Email already exists');
    } finally {
      setLoading(false);
    }
  };

  return (
    <SafeAreaView style={styles.container}>
      <View style={styles.header}>
        <Text style={styles.title}>Register</Text>
      </View>
      <TextInput
        style={styles.input}
        placeholder="Username"
        value={username}
        onChange={setUsername}
      />
      <TextInput
        style={styles.input}
        placeholder="Email"
        value={email}
        onChange={setEmail}
        keyboardType="email-address"
        autoCapitalize="none"
      />
      <TextInput
        style={styles.input}
        placeholder="Password"
        value={password}
        onChange={setPassword}
        secureTextEntry
      />
    </SafeAreaView>
  );
}

export default RegisterScreens;

```

3. Lupa Password

Tampilan lupa password digunakan ketika user lupa password dari email, pertama user akan memasukkan email terlebih dahulu. Kemudian sistem akan mengirim OTP ke email, inputkan OTP tersebut. User akan diminta membuat ulang password.

```

const ForgetPasswordScreenNavigationProp = ReactStackScreenProps<{forgetPassword: {navigation}: any}>;

const ForgetPasswordScreen = React.FC<{}> => {
  const [email, setEmail] = useState("");
  const [otp, setOtp] = useState("");
  const [otpSent, setOtpSent] = useState(false);
  const [navigation] = useNavigation<ForgetPasswordScreenNavigationProp>();

  const handleSendOTP = async () => {
    if (!email) {
      Alert.alert("Error", "Mohon masukan alamat email");
      return;
    }

    setOtpSent(true);

    try {
      const response = await axios.post(`http://192.168.0.111:1900/send-otp`, { email });
      Alert.alert("Success", "OTP berhasil dikirim ke email");
      navigation.navigate("VerifyOTP", { email });
    } catch (error) {
      console.error(error);
      Alert.alert("Error", "Failed to send OTP");
    }
  };

  const handleVerifyOTP = async () => {
    if (!otp) {
      Alert.alert("Error", "Masukan OTP");
      return;
    }

    const response = await axios.post(`http://192.168.0.111:1900/verify-otp`, { otp });
    if (response.data.success) {
      Alert.alert("Success", "OTP berhasil diverifikasi");
      navigation.navigate("ResetPassword");
    } else {
      Alert.alert("Error", "OTP tidak valid");
    }
  };
};

return (
  <SafeAreaView style={styles.container}>
    <View style={styles.header}>
      <Text style={styles.title}>Forgot Password?</Text>
    </View>
    <FormInput
      style={styles.input}
      placeholder="Email"
      value={email}
      onChangeText={setEmail}
      keyboardType="email-address"
      autoCapitalize="none"
      autoCorrect={false}
    />
    <TouchableOpacity style={styles.button} onPress={handleSendOTP}>
      <Text style={styles.buttonText}>Send OTP</Text>
      <ActivityIndicator size="small" color="black" style={styles.loading} />
    </TouchableOpacity>
  </SafeAreaView>
);
}

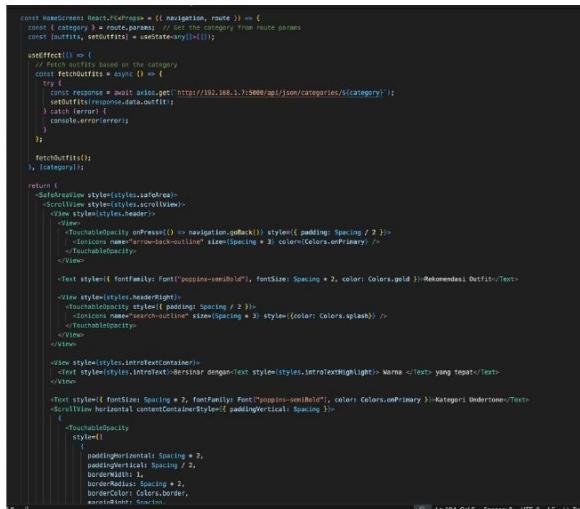
```

4. Deteksi

Halaman Deteksi dibuat untuk klasifikasi gambar dengan cara mengunggah foto atau mengambil foto secara langsung. Setelah foto diunggah, pengguna dapat menekan tombol "Deteksi Skin Tone" untuk mendeteksi warna skin tone atau tombol "Deteksi Warna" untuk mendeteksi warna.

5. Outfit

Setelah pengguna mengunggah atau mengambil gambar dan menekan tombol "Deteksi Skin Tone", mereka akan diarahkan ke halaman Outfit. Outfit yang ditampilkan berdasarkan hasil deteksi, yaitu Cool, Neutral, atau Warm.



```
const HomeScreen = React.FC<Props> = ({ navigation, route }) => {
  const { category } = route.params; // get the category from route params
  const [outfits, setOutfits] = useState([]);

  useEffect(() => {
    const fetchOutfits = async () => {
      try {
        const response = await axios.get(`http://192.168.1.7:5000/api/json/categories/${category}`);
        setOutfits(response.data.outfit);
      } catch (error) {
        console.error(error);
      }
    };
    fetchOutfits();
  }, [category]);
  return (
    <SafeAreaView style={styles.container}>
      <ScrollView style={styles.scrollView}>
        <View style={styles.header}>
          <TouchableOpacity style={styles.backIcon}>
            <Icon name="arrow-back-outline" size={Spacing * 2} color={Colors.onPrimary} />
          </TouchableOpacity>
        </View>
        <Text style={fontFamily: Font["poppins-semibold"], fontSize: Spacing * 2, color: Colors.gold}>Rekomendasi Outfit</Text>
        <View style={styles.headerRight}>
          <TouchableOpacity style={padding: Spacing / 2}>
            <Icon name="arrow-right-outline" size={Spacing * 3} style={{color: Colors.splash}} />
          </TouchableOpacity>
        </View>
        <View style={styles.insetTextContainer}>
          <Text style={fontStyle: 'italic', color: Colors.onPrimary}>Kategori: <Text style={color: Colors.splash}>Warna </Text> yang tepat</Text>
        </View>
        <Text style={fontStyle: 'italic', color: Colors.onPrimary}>Kategori: <Text style={color: Colors.splash}>Warna </Text> yang tepat</Text>
        <ScrollView horizontal contentContainerStyle={paddingVertical: Spacing}>
          {
            Touchableopacity
            <View style={paddingHorizontal: Spacing * 2, paddingVertical: Spacing / 2, borderRadius: Spacing * 2, border: 1, borderStyle: 'solid', borderColor: Colors.border, backgroundColor: Colors.onPrimary}>
              <Image style={borderRadius: Spacing * 2} source={outfits[0].image} />
              <Text style={fontStyle: 'italic', color: Colors.onPrimary}>Kategori: <Text style={color: Colors.splash}>Warna </Text> yang tepat</Text>
            </View>
            <View style={paddingHorizontal: Spacing * 2, paddingVertical: Spacing / 2, borderRadius: Spacing * 2, border: 1, borderStyle: 'solid', borderColor: Colors.border, backgroundColor: Colors.onPrimary}>
              <Image style={borderRadius: Spacing * 2} source={outfits[1].image} />
              <Text style={fontStyle: 'italic', color: Colors.onPrimary}>Kategori: <Text style={color: Colors.splash}>Warna </Text> yang tepat</Text>
            </View>
            <View style={paddingHorizontal: Spacing * 2, paddingVertical: Spacing / 2, borderRadius: Spacing * 2, border: 1, borderStyle: 'solid', borderColor: Colors.border, backgroundColor: Colors.onPrimary}>
              <Image style={borderRadius: Spacing * 2} source={outfits[2].image} />
              <Text style={fontStyle: 'italic', color: Colors.onPrimary}>Kategori: <Text style={color: Colors.splash}>Warna </Text> yang tepat</Text>
            </View>
          }
        </ScrollView>
      </ScrollView>
    </SafeAreaView>
  );
}
```

6. Detail Outfit

Pada halaman Detail Outfit, pengguna dapat melihat outfit yang ditampilkan berdasarkan hasil deteksi. Jika pengguna memilih outfit, mereka akan diarahkan ke halaman ini yang berisi foto outfit, detail kombinasi outfit, dan tiga palet kombinasi warna.

```

const useFetchDetails = (route, navigation) => {
  useEffect(() => {
    // Fetch product details based on id
    const fetchProductDetails = async () => {
      try {
        const response = await fetch(`http://192.168.1.7:5000/api/json/product/1/`);
        setProduct(response.data.detail[0]);
      } catch (error) {
        console.error(error);
      }
    };
    fetchProductDetails();
  }, [route]);
  if (!product) {
    return <Text> Loading...</Text>;
  }
  return (
    <SafeAreaView style={styles.safeArea}>
      <ScrollView style={paddingHorizontal: Spacing * 2}>
        <View style={flexDirection: 'row', justifyContent: 'space-between'}>
          <Touchableopacity onpress={() => navigation.goBack()} style={padding: Spacing / 2}>
            <Image style={margin: Spacing / 2, width: 15, height: 15} />
          <Text style={fontFamily: Font['poppins-semibold'], fontLetterSpacing: 2, color: Colors.onPrimary}>Detail</Text>
        </View>
        <Text style={padding: Spacing / 2}>
          <Text style={fontFamily: Font['poppins-semibold'], color: Colors.splash}>Product Info</Text>
        </Text>
        <Text style={color: Colors.onPrimary}>Name & Warna </Text>
        <View style={flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingVertical: Spacing}>
          <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Name</Text>
          <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Warna</Text>
        </View>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Description</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Price</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Stock</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Category</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Rating</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Reviews</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Share</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Add to Cart</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Buy Now</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Compare</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Share</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Add to Cart</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Buy Now</Text>
        <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Compare</Text>
      </ScrollView>
    </SafeAreaView>
  );
}

```

7. Detail Warna

Detail Warna menampilkan rincian rekommendasi warna dari hasil warna yang terdeteksi. Halaman ini akan menampilkan gambar warna yang cocok dan deskripsi mengapa warna tersebut cocok untuk dikombinasikan.

```

Rakuhus-7-E-commerce-screens-DetailedView > WarnaDetail > WarnaDetail > (useEffect) callback > (n) fetchWarnaDetails
1 const [idName] = route.params // Get the Name ID from route params
2 const [product] = useState[]
3
4 useEffect(() => {
5   const fetchWarnaDetails = async () => {
6     const response = await fetch(`http://192.168.1.7:5000/api/json/verno_detail/1/${idName}`);
7     setWarnaDetails(response.data.detail[0]);
8   }
9   fetchWarnaDetails();
10 }, [idName]);
11
12 if (!product) {
13   return <Text> Loading...</Text>;
14 }
15
16 return (
17   <SafeAreaView style={styles.safeArea}>
18     <ScrollView style={paddingHorizontal: Spacing * 2}>
19       <View style={flexDirection: 'row', justifyContent: 'space-between'}>
20         <Touchableopacity onpress={() => navigation.goBack()} style={padding: Spacing / 2}>
21           <Image style={margin: Spacing / 2, width: 15, height: 15} />
22         <Text style={fontFamily: Font['poppins-semibold'], fontLetterSpacing: 2, color: Colors.onPrimary}>Detail</Text>
23       </View>
24       <Text style={padding: Spacing / 2}>
25         <Text style={fontFamily: Font['poppins-semibold'], color: Colors.splash}>Product Info</Text>
26       </Text>
27       <Text style={color: Colors.onPrimary}>Name & Warna </Text>
28       <View style={flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingVertical: Spacing}>
29         <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Name</Text>
30         <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Warna</Text>
31       </View>
32       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Description</Text>
33       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Price</Text>
34       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Stock</Text>
35       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Category</Text>
36       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Rating</Text>
37       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Reviews</Text>
38       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Share</Text>
39       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Add to Cart</Text>
40       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Buy Now</Text>
41       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Compare</Text>
42       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Share</Text>
43       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Add to Cart</Text>
44       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Buy Now</Text>
45       <Text style={fontFamily: Font['poppins-semibold'], color: Colors.onPrimary}>Compare</Text>
46     </ScrollView>
47   </SafeAreaView>
48 )

```

Lampiran 5 Sertifikat HKI



LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Firly Aulia Azzahra	Jl. Cempaka No.48 RT/RW 10/01 Kec.Tegal Timur Kota Tegal Provinsi Jawa Tengah 52123, Tegal Timur, Tegal
2	Dwi Intan Af' idah, M.Kom	Desa Grinting RT/RW 003/001 Kec.Bulakamba Kab.Brebes Provinsi Jawa Tengah 52253, Bulakamba, Brebes
3	Ardi Susanto, S.Kom., M.Cs.	Jl. Banyuwangi RT 08 RW 01, Sumurpanggang Kec.Margadana Kota Tegal Provinsi Jawa Tengah 52143, Margadana, Tegal



Lampiran 6. Lembar Bimbingan



SARJANA TERAPAN TEKNIK INFORMATIKA POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Firly Aulia Azzahra
Nim : 20090064
No. Ponsel : 085725712476
Judul TA : Korelasi Warna Pakaian Berdasarkan Warna Kulit Menggunakan Metode K-Means Clustering dan YOLOv8 Berbasis Mobile
Dosen Pembimbing II : Dwi Intan Afidah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21 Maret 2024	Menjelaskan konsep aplikasi menggunakan Use Case	Tidak Ada	
2.	03 April 2024	Menjelaskan masih konsep aplikasi menggunakan Use case	Lansung membuat aplikasi	
3.	25 April 2024	Menampilkan model CNN	Menambah dataset atau mencari dataset yang memiliki kualitas gambar yang baik.	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
4.	16 Mei 2024	Menampilkan model CNN	Mencoba lagi, karena akurasi yang didapatkan masih sedikit	
5.	30 Mei 2024	Menampilkan model K-Means Clustering	Algoritma ini kurang sesuai untuk klasifikasi gambar, mengganti metode k-means clustering.	
6.	8 Juli 2024	Aplikasi	Menambah fitur matching warna	
7.	13 Juli 2024	Aplikasi	Tidak Ada, melanjutkan Manual Book dan Dokumen Teknis	
8.	17 Juli 2024	Laporan	Melanjutkan hingga selesai	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
3.	22 Juli 2024	Laporan	Acc	<i>Dwi Intan Afidah</i>

Tegal, Juli 2024
Dosen Pembimbing II


Dwi Intan Afidah, S.T., M.Kom.
NIP. 11.020.470



SARJANA TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN SKRIPSI

Nama : Firly Aulia Azzahra
NIM : 20090064
No. Ponsel : 085725712476
Judul Skripsi : Aplikasi Korelasi Warna Pakaian Berdasarkan Warna Kulit Menggunakan Metode K-Means Clustering dan CNN Berbasis Mobile
Dosen Pembimbing I : Ardi Susanto, S.Kom., M.Cs.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	3-4-24	- Model - Output	- Hasil dasar diperbaiki. - Output dg tampilan pakaian.	<i>Ary</i>
2.	29-5-24	Aplikasi	- Aplikasi dibuat sedangkan - Pembuatan rangangan model. CNN pertama	<i>Ary</i>
3.	2/6/2024	Model	Uji coba dg user dg SVM	<i>Ary</i>
4.	8/6/2024	Pembuatan Model	Pembuatan selanjutnya	<i>Ary</i>

R.	8 Juli 2024			
C.	13 Juli 2024			
F.	14 Juli 2024			
D.	17 Juli 2024.			

Tegal, 03 April 2024

Dosen Pembimbing I


Ardi Susanto S.Kom., M.Cs.
 NIPY. 03.021.490