

LAMPIRAN

Lampiran 1 Surat Kesediaan Pembimbing

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Naovi Magfiroh
NIM : 20090135
Program Studi : D IV Teknik Informatika

Pihak Kedua

Nama : Ir. Ginanjar Wiro Sasmito, M.Kom.
Status : Dosen
NIDN : 0613028601
Jabatan Fungsional : Lektor Kepala
Pangkat/Golongan : Penata Tingkat I/III D

Pada hari ini Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I 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.

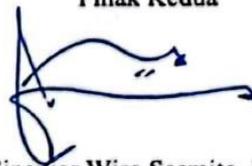
Tegal, 14 Maret 2024

Pihak Pertama



Naovi Magfiroh

Pihak Kedua



Ir. Ginanjar Wiro Sasmito, M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Naovi Magfiroh
NIM : 20090135
Program Studi : D IV Teknik Informatika

Pihak Kedua

Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Status : Dosen
NIDN : 0618119101
Jabatan Fungsional : Asisten Ahli
Pangkat/Golongan : Penata Muda Tingkat I/III B

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 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, 14 Maret 2024

Pihak Pertama



Naovi Magfiroh

Pihak Kedua



Hepatika Zidny Ilmadina, S.Pd., M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.P., M.Kom.

NIPY. 09.015.225

Lampiran 2 Surat Pengalihan Hak Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Naovi Magfiroh
Kewarganegaraan : Indonesia
Alamat : Jln. DR. Sarjito No. 28, RT 05 RW 03, Kelurahan Gandasuli, Kecamatan Brebes, Kabupaten Brebes
2. Nama : Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.
Kewarganegaraan : Indonesia
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba, Kabupaten Brebes
3. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jln. Kenanga Gang 1 No. 9, Kelurahan Mangkukusan, Kecamatan Tegal Timur, 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 “Aplikasi Deteksi Penyakit Kulit Menggunakan Arsitektur *Resnet50* Dengan Memanfaatkan Teknologi *Chatbot* Berbasis *Website*”. 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, 11 Juli 2024

Pemegang Hak Cipta

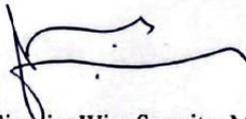


(Dr. Aldi Budi Riyanta, S.Si., M.T.)

Pencipta



(Naovi Magfiroh)



(Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng)



(Hepatika Zidny Ilmadina, S.Pd, M.Kom.)

Lampiran 3 Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Naovi Magfiroh
Kewarganegaraan : Indonesia
Alamat : Jln. DR. Sarjito No. 28, RT 05 RW 03, Kelurahan Gandasuli,
Kecamatan Brebes, Kabupaten Brebes
2. Nama : Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.
Kewarganegaraan : Indonesia
Alamat : Jln. Raya Kluwut Timur No. 24, Kecamatan Bulakamba,
Kabupaten Brebes
3. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jln. Kenanga Gang 1 No. 9, Kelurahan Mangkukusan,
Kecamatan Tegal Timur, Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Aplikasi Deteksi Penyakit Kuit Menggunakan Arsitektur *Resnet50*
Dengan Memanfaatkan Teknologi *Chatbot* Berbasis *Website*
 - 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 perkara 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, 11 Juli 2024



(Naovi Magfiroh)

Pemegang Hak Cipta *

(Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.)

Pemegang Hak Cipta *

(Hepatika Zidny Ilmadina, S.Pd, M.Kom.)

Pemegang Hak Cipta *

* Semua pemegang hak cipta agar menandatangani di atas materai.

MANUAL BOOK

Aplikasi Deteksi Penyakit Kulit Menggunakan Arsitektur
Resnet50 Dengan Memanfaatkan Teknologi Chatbot
Berbasis Website

DERMABOT

Penulis :

Naovi Magfiroh

Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng

Hepatika Zidny Ilmadina, S.Pd., M.Kom.

1. PENDAHULUAN

1.1. Tujuan Pembuatan Dokumen

Pembuatan dokumen manual book ini bertujuan untuk memberikan panduan yang jelas dan komprehensif tentang penggunaan aplikasi Dermabot, sehingga mempermudah pengguna dalam menjalankan aplikasi ini dengan efektif dan efisien.

1.2 Deskripsi Umum Sistem

1.1.1. Deskripsi Umum Aplikasi

Aplikasi Deteksi Penyakit Kulit (Dermabot) ini adalah sebuah *website* yang berfungsi untuk mendeteksi 10 jenis penyakit kulit. Sistem ini dapat digunakan oleh pengguna untuk melakukan deteksi melalui *real-time capture* atau *upload image* ke dalam sistem. Gambar tersebut akan diproses untuk diklasifikasikan ke dalam kategori penyakit yang sesuai, dan hasil klasifikasi ini akan ditampilkan pada halaman khusus yang berisi hasil klasifikasi dan rekomendasi obat yang tepat. Fitur *chatbot* pada aplikasi ini sangat bermanfaat bagi pengguna yang ingin mengetahui cara perawatan yang tepat sesuai dengan penyakit yang dialami. Selain itu, aplikasi ini dilengkapi dengan artikel yang berisi informasi tentang 10 jenis penyakit kulit. Hal ini memungkinkan pengguna untuk mendapatkan informasi yang akurat dan langkah-langkah perawatan yang disarankan tanpa harus berkonsultasi langsung dengan dokter.

1.1.2. Deskripsi Umum Kebutuhan Aplikasi

Deskripsi umum kebutuhan aplikasi yang akan diterapkan mencakup semua informasi teknis yang berfungsi sebagai panduan dalam pengembangan aplikasi Dermabot.

1.2. Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan penggunaan dari website Dermabot. Berikut adalah daftar informasi yang tersedia dalam buku panduan ini :

1. BAB I

Bab 1 ini memuat informasi terkait pendahuluan, yang mencakup tujuan pembuatan dokumen untuk memberikan panduan penggunaan aplikasi, deskripsi umum sistem serta kebutuhan aplikasi yang akan diimplementasikan, serta deskripsi dokumen yang menyajikan panduan lengkap penggunaan aplikasi deteksi penyakit kulit.

2. BAB II

Bab 2 ini memuat daftar perangkat yang diperlukan untuk menggunakan aplikasi, termasuk perangkat lunak dan juga perangkat keras yang dibutuhkan.

3. BAB III

Bab3 ini memuat penjelasan serta penggunaan fitur-fitur yang ada pada website Dermabot.

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak

Perangkat lunak yang dibutuhkan adalah sebagai berikut:

- Sistem Operasi: Windows 10 atau versi yang lebih baru
- *Web Browser: Google Chrome, Mozilla Firefox atau Microsoft Edge*
- Database: *MySQL*
- Bahasa Pemrograman: Python 3.9.13 atau versi lebih baru
- Library: *Tensorflow* untuk implementasi *Resnet50*, *Opencv* untuk pemrosesan gambar, dan library tambahan lainnya yang relevan dengan teknologi *chatbot*.
- Editor Kode: *Visual Studio Code*

2.2. Perangkat Keras

Perangkat keras yang dibutuhkan adalah sebagai berikut:

- Komputer atau Laptop
- *Mouse*
- *Keyboard*

3. MENU DAN CARA PENGGUNAAN

3.1. Struktur Menu

Struktur menu pada *website* Dermabot adalah sebagai berikut:

- Halaman Home
- Halaman Login
- Halaman Register
- Halaman Index
- Halaman Deteksi
- Halaman Realtime Capture
- Halaman Upload Image
- Halaman Chatbot
- Halaman Profil
- Halaman Ubah Pasword

3.2. Cara Penggunaan

Panduan penggunaan *website* Dermabot adalah sebagai berikut:

1. Buka Aplikasi DermaBot

Akses aplikasi DermaBot menggunakan *web* browser melalui URL yang telah disediakan. Saat pertama kali pengguna mengakses URL tersebut, pengguna akan diarahkan ke halaman *Home*.

2. Halaman *Home*

Pada halaman home, terdapat tombol "*Login*" yang dapat digunakan pengguna untuk mengakses aplikasi ini. Halaman *home* bisa dilihat pada Gambar 3.1.

DermaBot

Halo, Selamat Datang

Tingkatkan Kesehatan Kulit Anda dengan DermaBot

Aplikasi Pintar untuk Deteksi dan Perawatan Kulit yang Mudah!

Login



Gambar 3. 1 Halaman Home

3. Halaman Login

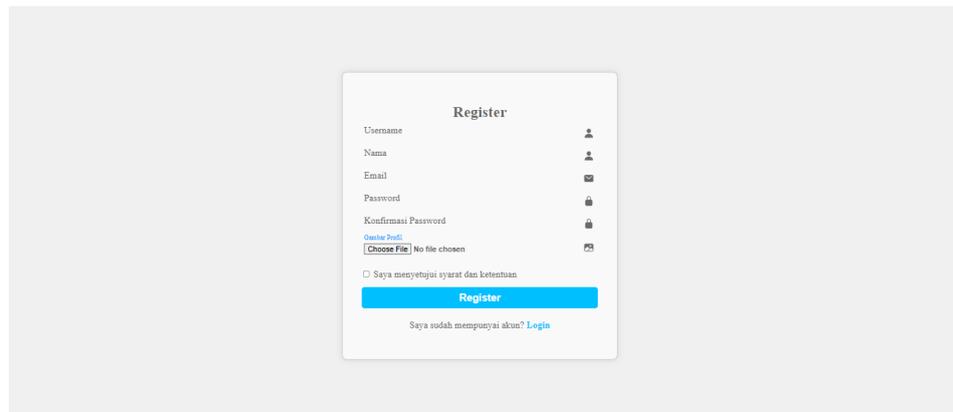
Pada halaman *login*, jika pengguna sudah memiliki akun, pengguna dapat langsung masuk dengan memasukkan data email dan *password*. Namun, jika pengguna belum memiliki akun, pengguna dapat mendaftar dengan mengklik tombol “*Register*”. Halaman *login* bisa dilihat pada Gambar 3.2.

A screenshot of a login form titled "Login". It features two input fields: "Email" and "Password", each with a small icon to its right. Below the fields is a checkbox labeled "Ingatkan Saya". A prominent blue "Login" button is centered below the checkbox. At the bottom of the form, there is a link that says "Belum mempunyai akun? Register".

Gambar 3. 2 Halaman Login

4. Halaman Register

Pada halaman *register*, pengguna yang belum memiliki akun bisa mendaftar dengan memasukkan data username, nama, email, *password*, konfirmasi *password* dan gambar profil. Setelah berhasil mendaftar, pengguna dapat mengklik tombol “*Login*” untuk mengakses aplikasi. Halaman *register* bisa dilihat pada Gambar 3.3.

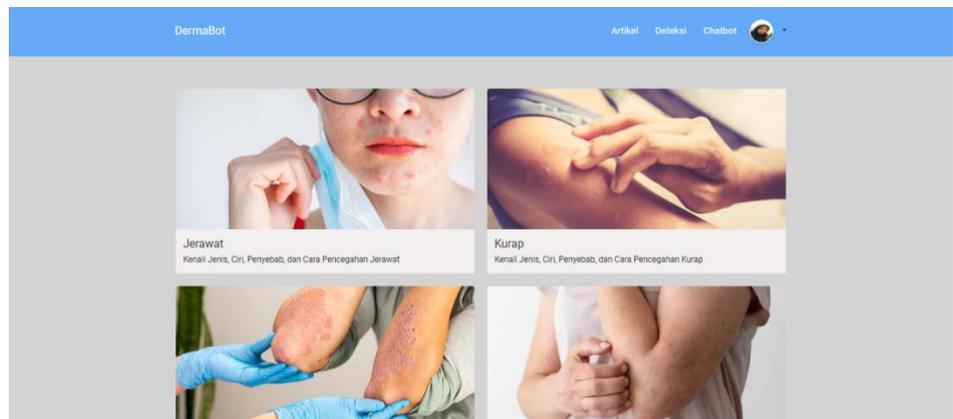


Gambar 3. 3 Halaman Register

5. Halaman Index

Pada halaman *index* yang merupakan halaman utama setelah pengguna berhasil *login*, akan ditampilkan artikel mengenai 10 jenis penyakit kulit.

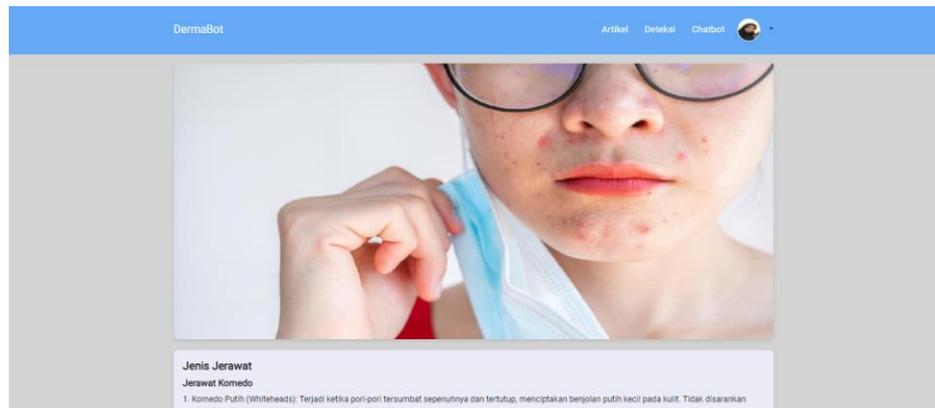
Halaman *index* bisa dilihat pada Gambar 3.4.



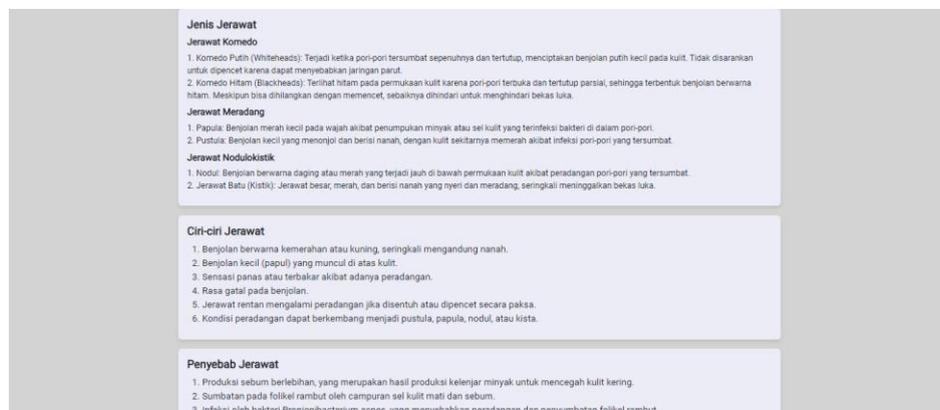
Gambar 3. 4 Halaman Index

a. Detail Artikel

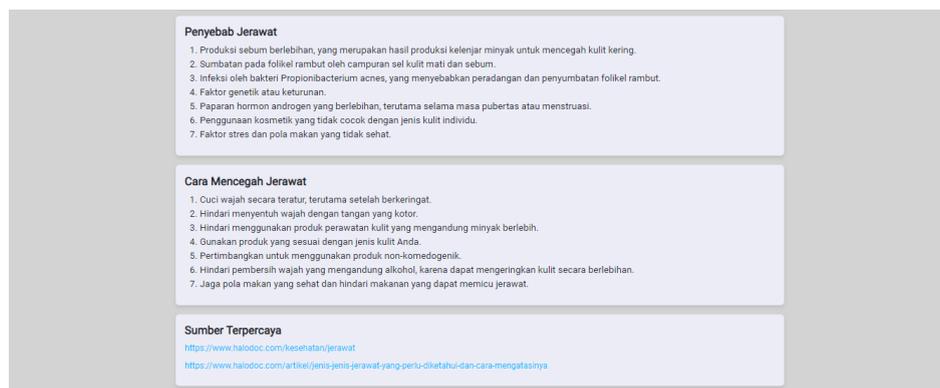
Setiap artikel memuat informasi terkait jenis penyakit, ciri-ciri, penyebab, cara pencegahan, dan sumber artikel terpercaya. Halaman detail *index* bisa dilihat pada Gambar 3.5 sampai 3.7.



Gambar 3. 5 Detail Artikel 1



Gambar 3. 6 Detail Artikel 2

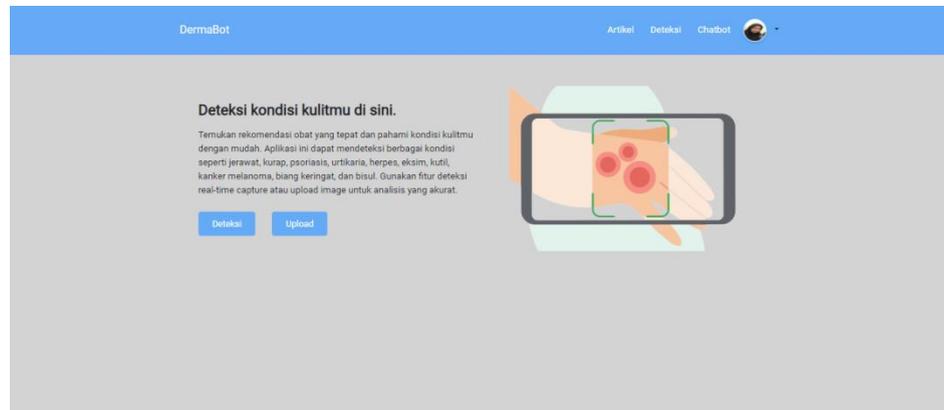


Gambar 3. 7 Detail Artikel 3

6. Halaman Deteksi

Pada halaman deteksi, terdapat dua cara yang dapat digunakan pengguna untuk mendeteksi penyakit kulit, yaitu dengan menggunakan deteksi *real-time capture* dan *upload image*. Pengguna dapat mengklik tombol “Deteksi” atau

“Upload” untuk melakukan deteksi penyakit sesuai dengan cara yang diinginkan. Halaman deteksi bisa dilihat pada Gambar 3.8.



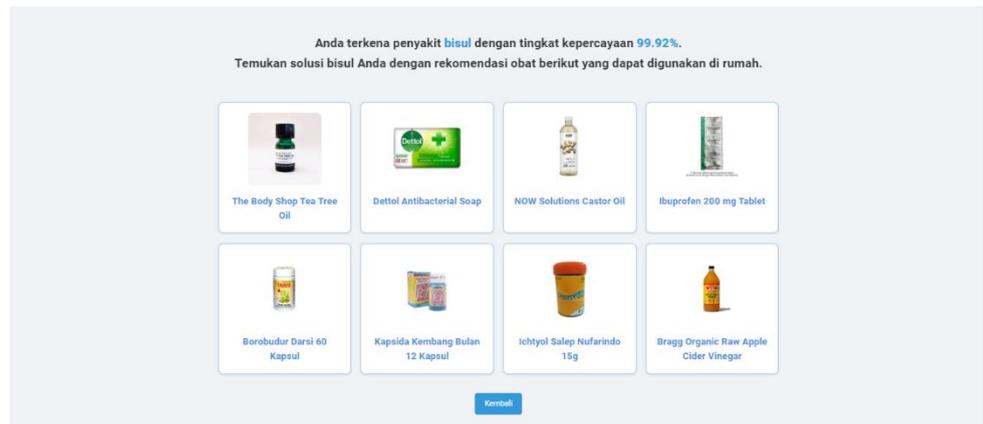
Gambar 3. 8 Halaman Deteksi

7. Halaman Real-Time Capture

Ketika pengguna mengklik tombol “Deteksi” pada halaman deteksi, aplikasi akan menampilkan kamera dengan *bounding box* yang menandai lokasi penyakit kulit. Pengguna dapat menggunakan tombol “Capture” di bawah kamera untuk mengambil gambar area penyakit yang sesuai. Gambar ini akan diproses untuk klasifikasi penyakit kulit. Gambar halaman deteksi *realtime-capture* bisa dilihat pada Gambar 3.9 dan hasil klasifikasi *realtime capture* beserta rekomendasi obat bisa dilihat pada Gambar 3.10.



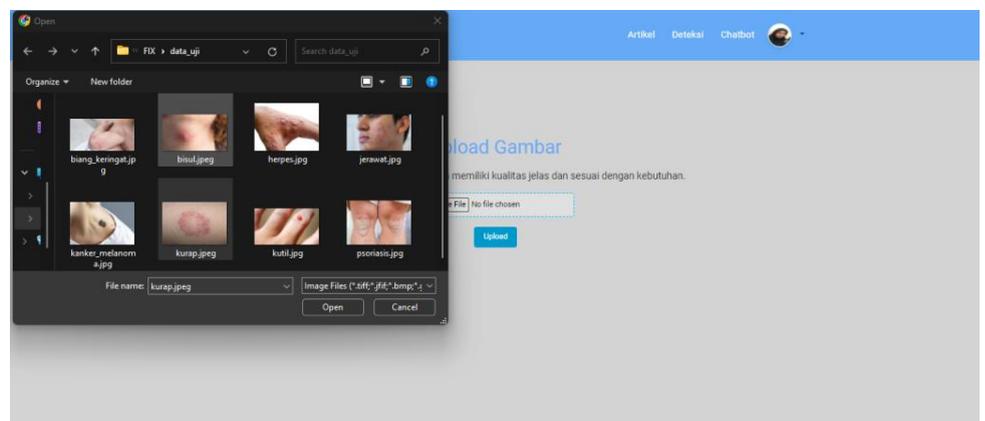
Gambar 3. 9 Halaman Deteksi Realtime Capture Menggunakan Kamera



Gambar 3. 10 Hasil Klasifikasi Realtime Capture dan Rekomendasi Obat

8. Deteksi Upload Image

Ketika pengguna mengklik tombol “*Upload*” pada halaman deteksi, aplikasi akan menampilkan halaman upload image. Pengguna dapat mengklik tombol “*Choose File*” untuk memilih gambar penyakit kulit yang ingin di deteksi. Proses memilih gambar dapat dilihat pada Gambar 3.11.



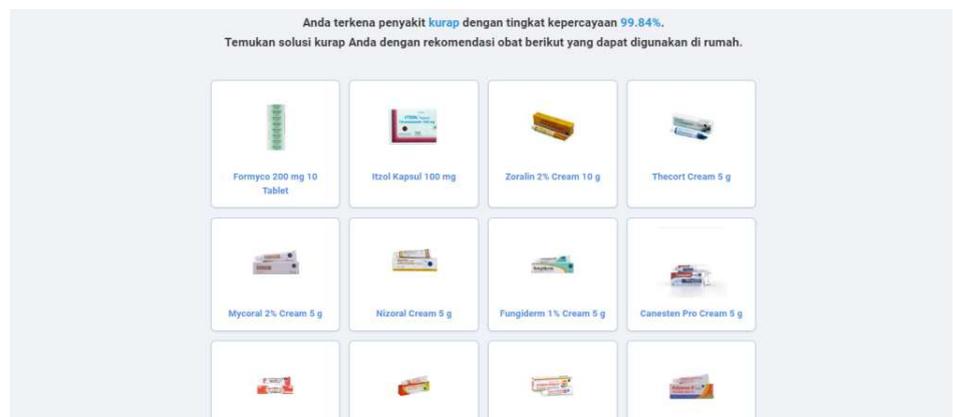
Gambar 3. 11 Memilih Gambar

Ketika gambar sudah terpilih, maka pengguna dapat mengklik tombol “*Upload*” untuk mengunggah gambar penyakit kulit yang ingin dideteksi. Gambar *upload image* bisa dilihat pada Gambar 3.12.



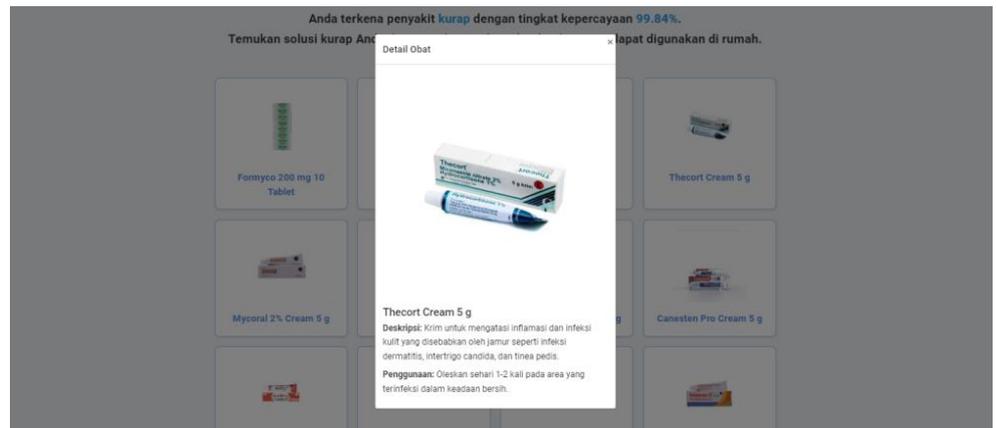
Gambar 3. 12 Upload Image

Setelah itu, aplikasi akan menampilkan hasil klasifikasi penyakit kulit beserta rekomendasi obat yang sesuai. Informasi ini akan membantu pengguna memahami jenis penyakit kulit yang terdeteksi dan langkah pengobatan yang direkomendasikan. Gambar hasil klasifikasi *upload image* dan rekomendasi obat bisa dilihat pada Gambar 3.13.



Gambar 3. 13 Hasil Klasifikasi Upload Image dan Rekomendasi Obat

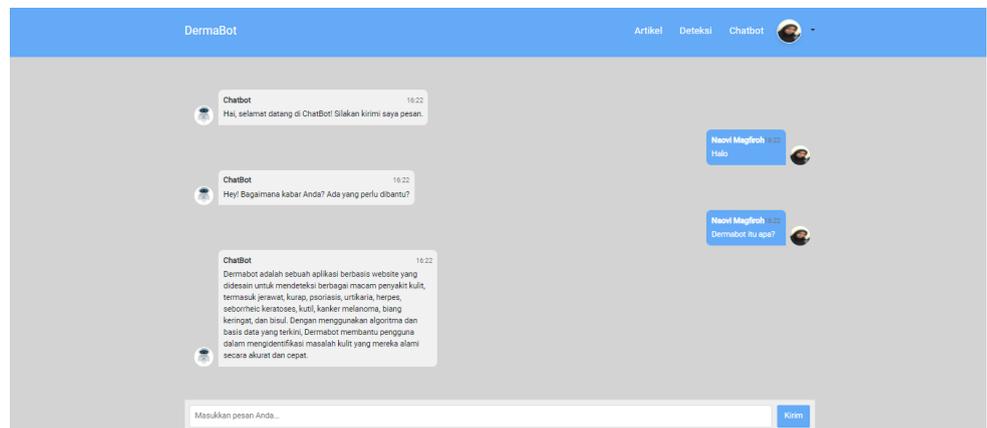
Rekomendasi obat tersebut juga dilengkapi dengan informasi detail yang mencakup deskripsi dan cara penggunaan obat. Gambar detail obat bisa dilihat pada Gambar 3.14.



Gambar 3. 14 Detail Obat

9. Halaman *Chatbot*

Pada halaman *chatbot*, pengguna dapat mengajukan berbagai pertanyaan terkait penyakit kulit. *Chatbot* ini dirancang untuk memberikan informasi mengenai jenis-jenis penyakit kulit, ciri-ciri masing-masing penyakit, penyebabnya, serta cara pencegahannya. Pengguna dapat berinteraksi dengan *chatbot* untuk mendapatkan jawaban yang relevan dan cepat tentang penyakit kulit yang ingin diketahui. Halaman *chatbot* bisa dilihat pada Gambar 3.15.

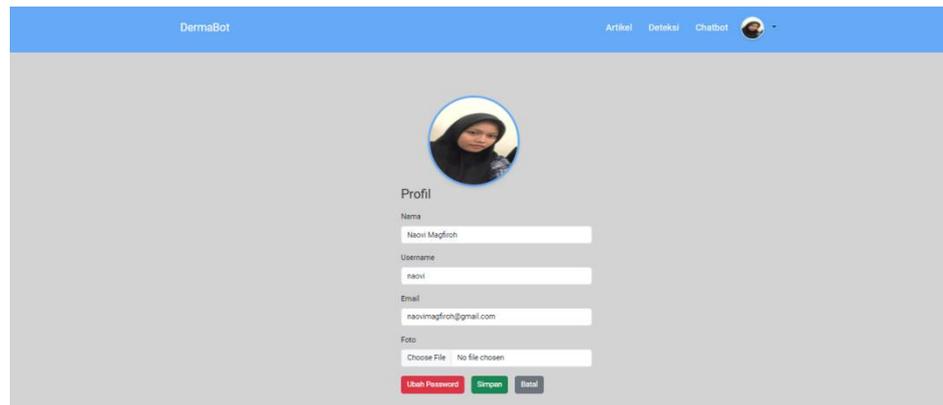


Gambar 3. 15 Halaman Chatbot

10. Halaman Profil

Pada halaman profil, pengguna dapat melihat data yang telah didaftarkan saat registrasi seperti nama, *username*, dan email. Di halaman ini, pengguna juga dapat mengubah data profil dan mengganti gambar profil. Setelah melakukan perubahan data, pengguna cukup mengklik tombol “Simpan”

untuk memperbarui informasi. Selain itu, terdapat tombol “Ubah Password” bagi pengguna yang ingin merubah *password*. Jika pengguna memutuskan untuk tidak melanjutkan perubahan, pengguna dapat membatalkannya dengan mengklik tombol “Batal”. Halaman profil bisa dilihat pada Gambar 3.16.



Gambar 3. 16 Halaman Profil

11. Halaman Ubah Password

Pada halaman ubah *password*, pengguna dapat melakukan perubahan dengan memasukkan *password* lama, *password* baru, dan mengkonfirmasi *password* baru. Setelah itu, pengguna dapat menyimpan perubahan dengan mengklik tombol “Ubah Password”. Jika pengguna memutuskan untuk tidak melanjutkan perubahan, pengguna dapat membatalkannya dengan mengklik tombol “Batal”. Halaman ubah *password* bisa dilihat pada Gambar 3.17.



Gambar 3. 17 Halaman Ubah Password

DOKUMEN TEKNIKAL

Aplikasi Deteksi Penyakit Kulit Menggunakan Arsitektur
Resnet50 Dengan Memanfaatkan Teknologi Chatbot
Berbasis Website

DERMABOT

Penulis

Naovi Magfiroh

Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.

Hepatika Zidny Ilmadina, S.Pd., M.Kom.

Profil

Aplikasi Dermabot adalah sebuah platform yang dirancang untuk mendeteksi penyakit kulit secara otomatis. Aplikasi ini memungkinkan pengguna untuk mendapatkan diagnosis awal penyakit kulit melalui deteksi *real-time* menggunakan kamera perangkat atau dengan mengunggah gambar. Dermabot juga dilengkapi dengan *chatbot* yang dapat memberikan informasi dan saran mengenai berbagai jenis penyakit kulit. Fitur lain termasuk halaman profil untuk memperbarui data pengguna dan halaman artikel yang memberikan pengetahuan terkait penyakit kulit.

Latar Belakang

Kulit merupakan organ yang sangat penting bagi manusia karena terletak di bagian luar tubuh dan rentan terhadap rangsangan seperti sentuhan, nyeri, dan pengaruh luar lainnya. Kulit yang tidak dijaga kesehatannya dapat menimbulkan berbagai macam penyakit kulit. Penyakit kulit juga dikenal sebagai penyakit yang umum terjadi di negara tropis seperti Indonesia dan dapat menyerang siapa saja dan bagian tubuh mana saja. Beberapa jenis penyakit kulit di antaranya adalah jerawat, kurap, psoriasis, urtikaria, herpes, seborrheic keratoses, kutil, kanker melanoma, biang keringat, dan bisul.

Kehadiran penyakit kulit memberikan dampak negatif seperti, menyebabkan sensasi gatal yang sangat intens, hal tersebut membuat penderita untuk menggaruk, mengakibatkan luka, dan berpotensi menimbulkan infeksi. Selain itu, dampak emosional seperti perasaan ketidaknyamanan, kurang percaya diri, dan rasa malu, bahkan bisa mengakibatkan gangguan psikologis yang lebih serius seperti cita diri yang negatif. Oleh karena itu, penyakit kulit perlu didiagnosa secara cepat dan akurat. Namun, proses diagnosa penyakit kulit seringkali mengalami kesulitan karena beberapa faktor, antara lain terbatasnya jumlah tenaga medis, tingginya biaya, dan kesulitan akses layanan kesehatan di daerah terpencil. Dengan demikian, untuk mengatasi permasalahan kesehatan kulit diperlukan

solusi yang mampu memberikan diagnosa awal penyakit kulit secara cepat dan akurat.

Seiring dengan perkembangan teknologi, penggunaan kecerdasan buatan (AI) semakin meluas, terutama dalam bidang *computer vision*. *Computer vision* adalah salah satu cabang AI yang paling berkembang pesat. Cabang ilmu ini mempelajari bagaimana komputer dapat mengenali dan memahami objek yang diamati dalam lingkungan visual. *Computer Vision* menggabungkan berbagai metode, termasuk deteksi objek dan klasifikasi gambar sebagai dua pendekatan utama. Teknologi ini dalam pemrograman komputer sangat terkait dengan pengolahan citra dan visi komputer. Fokusnya adalah mengidentifikasi objek dalam citra digital, melibatkan pengenalan warna dan bentuk objek dalam gambar. Klasifikasi citra merupakan proses klasifikasi citra yang mengelompokkan objek berdasarkan kelas tertentu. Oleh karena itu, kita dapat dengan mudah mengenali suatu objek.

Kemajuan teknologi telah mempermudah pembuatan *Haar Cascade Classifier* secara mandiri yang digunakan untuk pengenalan objek berbasis fitur. Berkat teknologi modern, *Haar Cascade Classifier* kini lebih mudah dan efisien untuk dikembangkan. Di sisi lain, algoritma CNN seperti arsitektur *ResNet50* dapat mengklasifikasikan gambar dan mendeteksi objek dengan sangat akurat, menjadikannya pilihan utama di bidang *computer vision*. Melalui *transfer learning*, *ResNet50* dapat memanfaatkan pengetahuan dari kumpulan data berskala besar sebelumnya untuk meningkatkan kinerja dan efisiensi untuk tugas klasifikasi tertentu, seperti deteksi penyakit kulit.

Mengingat besarnya potensi teknologi dan pentingnya diagnosis penyakit kulit, maka perlu dikembangkan aplikasi yang cepat dan akurat. Aplikasi ini bekerja dalam dua cara yaitu deteksi *real-time capture* dan *upload image*. Deteksi *real-time capture* menggunakan *Haar Cascade Classifier* yang dikembangkan khusus untuk penyakit kulit, mendeteksi area kulit yang bermasalah melalui kamera perangkat. Pengguna dapat mengklik tombol "*Capture*" ketika *bounding*

box sudah tepat berada di area kulit yang bermasalah. Sedangkan, untuk *upload image* pengguna dapat mengunggah gambar area kulit yang bermasalah. Hasil deteksi dari kedua metode tersebut dikirim ke server, dimana datanya diproses menggunakan algoritma CNN yang telah dilatih sebelumnya. Server kemudian menganalisis gambar dan membuat diagnosis berdasarkan informasi yang diterima. Hasil diagnosa dikirim kembali ke aplikasi untuk ditampilkan kepada pengguna. Aplikasi ini juga memberikan rekomendasi obat tergantung kondisi kulit yang didiagnosis serta memiliki fitur *chatbot* yang berinteraksi dengan pengguna serta memberikan tips perawatan kulit. Dengan adanya aplikasi ini, diharapkan para pengguna tidak usah kebingungan lagi ketika ingin mengetahui penyakit kulit tanpa harus berkunjung ke dokter. Dengan adanya aplikasi ini, diharapkan masyarakat dapat mendeteksi penyakit kulit tanpa konsultasi langsung ke dokter kulit.

Manfaat

Adapun manfaaat yang diperoleh dari aplikasi ini adalah sebagai berikut:

1. Memudahkan masyarakat untuk mendapatkan diagnosis awal penyakit kulit tanpa perlu berkunjung ke dokter, terutama di daerah terpencil dengan akses kesehatan yang terbatas.
2. Meningkatkan kesadaran masyarakat tentang pentingnya menjaga kesehatan kulit dan mengenali gejala awal penyakit kulit

Spesifikasi

1. Platform
 - Aplikasi berbasis *website*, dapat di akases melalui *Google Crome*, *Mozilla Firefox* dan *Microsoft Edge*.
2. Backend
 - Bahasa Pemrogramans: Python 3.9.13 atau versi lebih baru
 - Framework: Flask(Python) untuk pengelolaan server dan API
 - Database: MySQL menggunakan XAMPP untuk menyimpan data

pengguna pada saat login dan register

- Model *Haar Cascade Classifier* dan Model *Resnet50*: Untuk deteksi *realtime capture* dan *upload image* serta memberikan hasil diagnosis
- Model *Chatbot*: Untuk memproses input pengguna, menghasilkan respon yang sesuai.

3. Frontend

- Antarmuka *Website*: HTML, CSS dan JavaScript
- Libraries: Bootstrap untuk desain

4. Source Code

Berikut adalah source code yang digunakan dalam aplikasi Dermabot ini :

- Kode Pembuatan *Haar Cascade Classifier*

Dalam proses pembuatan cascade menggunakan platform Cascade Trainer GUI, setelah mendapatkan file XML untuk setiap jenis penyakit, file XML tersebut kemudian diproses menggunakan kode untuk mengatur parameter-parameter dalam cascade. Salah satunya adalah *scaleFactor*, yang menentukan seberapa banyak gambar akan diperkecil atau diperbesar pada setiap iterasi. Parameter *minNeighbors* digunakan untuk mengontrol jumlah minimum tetangga yang harus dimiliki setiap kotak persegi agar dianggap sebagai deteksi valid. Selain itu, parameter *minSize* digunakan untuk menentukan ukuran minimum objek, sehingga objek yang lebih kecil dari ukuran ini akan diabaikan dalam proses deteksi.

```
def detect_skin_disease(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (5, 5), 0)

    for cascade in cascades:
        if cascade.empty():
            continue
        detected_objects = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=20, minSize=(50, 50))
        if len(detected_objects) > 0:
            x, y, w, h = detected_objects[0]
            roi = image[y:y+h, x:x+w]
            cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
            return roi, True, (x, y, w, h)
    return image, False, None
```

- Kode Pembuatan Model *Transfer Learning*

Dalam pembuatan model transfer learning ini, terdapat beberapa tahap penting. Dimulai dari pengumpulan data gambar untuk 10 jenis penyakit kulit. Setelah gambar diperoleh, dilakukan *pre-processing* data dengan membagi data menjadi 70% untuk training, 15% untuk validasi, dan 15% untuk testing. Selanjutnya, dilakukan augmentasi data menggunakan ImageDataGenerator dengan berbagai teknik seperti rotasi, pergeseran horizontal dan vertikal, pemotongan, zoom, flip horizontal dan vertikal, serta variasi kecerahan.

```
train_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input,  
    rotation_range=30,  
    width_shift_range=0.3,  
    height_shift_range=0.3,  
    shear_range=0.3,  
    zoom_range=0.3,  
    horizontal_flip=True,  
    vertical_flip=True,  
    brightness_range=[0.2, 1.0],  
    fill_mode='nearest'  
)
```

Setelah konfigurasi ImageDataGenerator, data diproses menggunakan metode `flow_from_directory` untuk mengambil gambar dari direktori yang telah dibagi, dengan menentukan ukuran target gambar sesuai dengan kebutuhan *ResNet50* dan *batch size* yang sesuai.

```
train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(224, 224),  
    batch_size=64,  
    class_mode='categorical'  
)  
  
validation_generator = validation_datagen.flow_from_directory(  
    validation_dir,  
    target_size=(224, 224),  
    batch_size=64,  
    class_mode='categorical'  
)  
  
test_generator = test_datagen.flow_from_directory(  
    test_dir,  
    target_size=(224, 224),  
    batch_size=64,  
    class_mode='categorical'  
)
```

Setelah tahap pemrosesan gambar selesai, langkah berikutnya adalah membangun dan melatih model menggunakan arsitektur *ResNet50* yang sudah pre-trained. Model *ResNet50* dimuat tanpa lapisan akhir (`include_top=False`) agar dapat ditambahkan lapisan kustom untuk klasifikasi 10 kelas pada dataset penyakit kulit.

```
[ ] base_model = ResNet50(weights='imagenet', include_top=False)
↳ Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [-----] - 1s 0us/step
```

Langkah pertama adalah menambahkan lapisan-lapisan akhir pada model. Output dari *ResNet50* diteruskan ke lapisan *GlobalAveragePooling2D*, yang kemudian dihubungkan ke lapisan *Dense* dengan fungsi aktivasi *ReLU*. Untuk mencegah *overfitting*, lapisan *Dropout* ditambahkan setelah lapisan *Dense*. Sehingga, lapisan output dengan fungsi aktivasi *softmax* ditambahkan untuk menghasilkan prediksi kelas.

```
▶ x = base_model.output
  x = GlobalAveragePooling2D()(x)
  x = Dense(1024, activation='relu')(x)
  x = Dropout(0.5)(x)
  predictions = Dense(10, activation='softmax')(x)
```

Model akhir kemudian dibentuk dengan mendefinisikan *input* dan *output* dari model. Pada tahap awal *training*, semua lapisan pada *base model (ResNet50)* dibekukan agar bobot-bobot *pre-trained* tidak diubah. Model kemudian di-*compile* dengan *optimizer Adam*, menggunakan *loss function categorical_crossentropy* dan metrik akurasi.

```
[ ] # Bentuk model akhir
    model = Model(inputs=base_model.input, outputs=predictions)

[ ] # Bekukan lapisan awal
    for layer in base_model.layers:
        layer.trainable = False

[ ] # Compile model
    model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

Selanjutnya, beberapa *callbacks* didefinisikan untuk membantu proses *training*. *ModelCheckpoint* digunakan untuk menyimpan model terbaik

berdasarkan nilai minimal dari *val_loss*. *EarlyStopping* digunakan untuk menghentikan *training* jika *val_loss* tidak menunjukkan perbaikan setelah beberapa *epoch*. *ReduceLRonPlateau* digunakan untuk mengurangi *learning rate* jika *val_loss* tidak menunjukkan perbaikan setelah beberapa *epoch*.

```
[ ] # Definisikan callbacks
checkpoint = ModelCheckpoint('classification_model_resnet50_best.h5', monitor='val_loss', save_best_only=True, mode='min')
early_stopping = EarlyStopping(monitor='val_loss', patience=10, mode='min')
reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-7)
```

Proses *training* model dilakukan dengan memanggil metode *fit* pada objek model, menggunakan *train_generator* untuk data *training* dan *validation_generator* untuk data validasi. *Training* dilakukan selama beberapa *epoch* dengan jumlah langkah per *epoch* dihitung berdasarkan jumlah sampel dalam *train_generator* dibagi dengan *batch size*. *Callbacks* yang telah didefinisikan juga diterapkan selama proses *training* untuk memantau dan mengatur penyesuaian model secara otomatis.

```
# Latih model sebelum fine-tuning
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    epochs=40,
    callbacks=[checkpoint, early_stopping, reduce_lr]
)
```

Perolehan akurasi sebelum di *fine-tuning*

```
Epoch 36/40
10/10 [=====] - 155s 16s/step - loss: 0.3751 - accuracy: 0.8850 - val_loss: 0.3277 - val_accuracy: 0.8906 - lr: 2.0000e-05
```

Setelah *training* awal, tahap *fine-tuning* dilakukan untuk lebih mengoptimalkan performa model. Dalam tahap ini, beberapa lapisan awal dari *base model* tetap dibekukan, sementara lapisan-lapisan lainnya ditraining ulang dengan *learning rate* yang lebih rendah. Model *ResNet50* yang digunakan memiliki total 175 lapisan. Pada proses *fine-tuning* ini, sebagian lapisan awal dibekukan untuk mempertahankan fitur-fitur umum yang telah dipelajari dari dataset *ImageNet*, sehingga model dapat lebih

fokus pada fitur-fitur spesifik dari dataset klasifikasi penyakit kulit yang digunakan.

```
# Fine-tuning: Buka beberapa lapisan awal dan latih ulang dengan learning rate yang lebih rendah
for layer in base_model.layers[:100]:
    layer.trainable = False
for layer in base_model.layers[100:]:
    layer.trainable = True
```

Proses *fine-tuning* dilakukan dengan mengatur ulang model menggunakan *learning rate* yang lebih rendah. Model kemudian dilatih ulang selama beberapa *epoch* menggunakan data *training* yang sama, dengan validasi pada setiap *epoch* untuk memantau performa.

```
model.compile(optimizer=Adam(learning_rate=1e-5), loss='categorical_crossentropy', metrics=['accuracy'])

# Latih model sesudah fine-tuning
history_fine_tune = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    epochs=20,
    callbacks=[checkpoint, early_stopping, reduce_lr]
)
```

Perolehan akurasi sesudah dilakukan *fine-tuning*

```
Epoch 20/20
10/10 [=====] - 272s 27s/step - loss: 0.3107 - accuracy: 0.9010 - val_loss: 0.3434 - val_accuracy: 0.8906 - lr: 4.0000e-07
```

Model yang telah melalui proses *fine-tuning* disimpan dalam format h5 agar dapat diintegrasikan dengan *website flask*.

- Kode Pembuatan Model Chatbot Menggunakan Deep Learning

Dalam pembuatan model *chatbot* ini, pertama kali yang dibuat adalah pembuatan file *json* berisi *intent*, *pattern*, dan *response* yang akan digunakan oleh *chatbot*.

```

{"intents": [
  {
    "tag": "sapa",
    "patterns": ["hallo", "hei", "hai", "halo", "hi", "hii", "hiii", "helo", "hey", "yo"],
    "responses": ["Halo! Ada yang bisa dibantu?", "Apakah ada yang bisa saya bantu?", "Hai! Bagaimana saya bisa membantu Anda hari ini?", "Halo! Apa yang bisa saya bantu?", "Selamat datang! Ada yang bisa saya bantu?", "Hai! Butuh bantuan dengan sesuatu?", "Hi! Ada yang bisa saya lakukan untuk Anda?", "Hey! Bagaimana kabar Anda? Ada yang perlu dibantu?", "Apa kabar? Ada yang bisa saya bantu?"]
  },
  {
    "tag": "nama",
    "patterns": ["siapa namamu", "nama kamu siapa", "kamu siapa", "ini siapa"],
    "responses": ["Saya adalah DermaBot, aplikasi pintar untuk deteksi dan perawatan kulit yang mudah!"]
  },
  {
    "tag": "tanya",
    "patterns": ["aku mau tanya", "saya mau bertanya", "aku bertanya", "saya bertanya", "mau tanya dong", "boleh bertanya?", "mau tanya nih", "saya ada pertanyaan", "bisa jawab pertanyaanku?", "saya ingin tanya sesuatu", "ada yang ingin saya tanyakan", "saya ingin konsultasi", "bolehkah saya tanya?", "saya mau tanya tentang", "ada yang ingin saya tanyakan"],
    "responses": ["Silakan! mau bertanya tentang apa?", "Tentu, ada yang ingin kamu tanyakan?", "Ya, apa yang ingin kamu ketahui?", "Pasti, silakan ajukan pertanyaamu!", "Tentu saja, ada yang ingin kamu tanyakan?", "Iya, ada yang bisa saya bantu?", "Tentu, saya siap menjawab pertanyaamu.", "Ya, silakan sampaikan pertanyaamu.", "Silakan, ada yang ingin kamu tanyakan?", "Tentu, saya akan berusaha menjawab pertanyaamu sebaik mungkin.", "Ya, saya siap menjawab pertanyaamu.", "Tentu saja, ada yang ingin kamu ketahui?", "Iya, ada yang bisa saya bantu?", "Pasti, silakan ajukan pertanyaamu!", "Tentu saja, apa yang ingin kamu tanyakan?"]
  },
  {
    "tag": "dermatobot",
    "patterns": ["aplikasi dermatob itu apa?", "apa itu dermatob?", "aplikasi dermatob itu apa?", "ini aplikasi apa?"],
    "responses": ["Dermatobot adalah sebuah aplikasi berbasis website yang didesain untuk mendeteksi berbagai macam penyakit kulit, termasuk jerawat, kutap, psoriasis, urtikaria, herpes, seborrheic keratosis, kutil, kanker melanoma, biang keringat, dan bisul."]
  }
],
}

```

Setelah memperoleh data *JSON*, langkah berikutnya adalah melakukan *pre-processing* yang dimulai dengan *tokenisasi* menggunakan *nlk.word_tokenize*. Proses ini memisahkan teks berdasarkan spasi dan tanda baca untuk menghasilkan kata-kata individual dari setiap kalimat.

```

lemmatizer = WordNetLemmatizer()

intents = json.loads(open('/content/drive/MyDrive/skripsi/chatbot_fix_banget_acc/chatbot.json').read())

words = []
classes = []
documents = []
ignore_letters = ['?', '!', ',', '.', ':']

for intent in intents['intents']:
    for pattern in intent['patterns']:
        word_list = nltk.word_tokenize(pattern)
        words.extend(word_list)
        documents.append((word_list,intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])

```

Setelah tokenisasi, kata-kata dihasilkan melalui proses lemmatisasi menggunakan *WordNetLemmatizer* dari *NLTK* untuk mengubah kata-kata ke bentuk dasarnya. Kata-kata yang tidak relevan, seperti tanda baca, diabaikan. Hasil lemmatisasi kemudian disimpan dalam file *words.pkl* dan *classes.pkl* menggunakan *pickle* untuk tahap berikutnya.

```

words = [lemmatizer.lemmatize(word) for word in words if word not in ignore_letters]
words = sorted(set(words))

classes = sorted(set(classes))

pickle.dump(words, open('words.pkl', 'wb'))
pickle.dump(classes, open('classes.pkl', 'wb'))

```

Metode *Bag of Words (BoW)* digunakan untuk membuat representasi numerik dari teks. Setiap kata dihitung kemunculannya dan diberi nilai 1 jika ada dalam kantung kata, atau 0 jika tidak ada. Data yang diolah kemudian diacak dan diubah menjadi *array numpy* untuk digunakan dalam pelatihan model *chatbot*.

```
training = []
output_empty = [0] * len(classes)

for document in documents:
    bag = []
    word_patterns = document[0]
    word_patterns = [lemmatizer.lemmatize(word.lower()) for word in word_patterns]
    for word in words:
        bag.append(1 if word in word_patterns else bag.append(0))

    output_row = list(output_empty)
    output_row[classes.index(document[1])] = 1
    training.append([bag, output_row])

[ ] #DATA YANG TELAH DI OLAH DIJADIKAN NUMPY ARRAY
random.shuffle(training)
training = np.array(training, dtype=object)

train_x = list(training[:, 0])
train_y = list(training[:, 1])
```

Proses *training* model dimulai dengan mendefinisikan arsitektur menggunakan model *Sequential*. Ini melibatkan penambahan lapisan *Dense* dengan aktivasi *ReLU* sebagai *input layer* untuk menerima vektor *Bag of Words*. Untuk mengurangi *overfitting*, ditambahkan lapisan *Dropout*. *Hidden layers* terdiri dari lapisan *Dense* dan *Dropout* yang diulang. *Output layer* memiliki neuron sesuai jumlah kelas *intents*, dengan aktivasi *softmax* untuk probabilitas kelas.

```
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

#sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
sgd = SGD(learning_rate=0.01, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5)
```

Perolehan akurasi model *chatbot*

Epoch 200/200
52/52 [=====] - 0s 4ms/step - loss: 0.3555 - accuracy: 0.8677

Model *chatbot* yang telah dilatih disimpan dalam format .h5 untuk memudahkan integrasi dengan aplikasi web berbasis Flask.

- Kode Halaman Home

Kode ini digunakan untuk menampilkan halaman *home* yang merupakan tampilan awal dari aplikasi, dimana terdapat tombol untuk *login*.

```
<div class="navbar">
  <div class="text">
    <div class="dermabot">DermaBot</div>
    <h1 class="selamat">Halo, Selamat Datang</h1>
    <h1 class="tingkatkan">Tingkatkan Kesehatan Kulit Anda dengan DermaBot</h1>
    <h1 class="aplikasi">Aplikasi Pintar untuk Deteksi dan Perawatan Kulit yang Mudah!</h1>
    <button class="button-login" onclick="window.location.href='login';">Login</button>
  </div>
  <div class="gambar">
    
  </div>
</div>
```

- Kode Halaman Login

Kode ini berfungsi untuk menampilkan halaman *login*, tempat dimana pengguna dapat memasukkan email dan *password* untuk masuk jika sudah memiliki akun.

```
<div class="wrapper">
  <div class="form-box login">
    <h2>Login</h2>
    <form action="/login" method="POST">
      <div class="input-box">
        <span class="icon"><ion-icon name="mail"></ion-icon></span>
        <input type="email" name="email" required>
        <label>Email</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
        <input type="password" name="password" required>
        <label>Password</label>
      </div>
      <div class="remember-forgot">
        <label><input type="checkbox">
          Ingatkan Saya</label>
      </div>
      <button type="submit" class="btn">Login</button>
      <div class="login-register">
        <p>Belum mempunyai akun?<a href="register" class="register-link">Register</a></p>
      </div>
    </form>
  </div>
</div>
```

- Kode Halaman Register

Kode ini digunakan untuk menampilkan halaman *register*, di mana pengguna dapat melakukan pendaftaran dengan mengisi data seperti username, nama, email, *password*, konfirmasi *password*, dan gambar profil.

```

<div class="wrapper">
  <div class="form-box register">
    <h2>Register</h2>
    <form action="/register" method="POST" enctype="multipart/form-data">
      <div class="input-box">
        <span class="icon"><ion-icon name="person"></ion-icon></span>
        <input type="text" name="username" required>
        <label>Username</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="person"></ion-icon></span>
        <input type="text" name="fullname" required>
        <label>Nama</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="mail"></ion-icon></span>
        <input type="email" name="email" required>
        <label>Email</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
        <input type="password" name="password" required>
        <label>Password</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
        <input type="password" name="confirm_password" required>
        <label>Konfirmasi Password</label>
      </div>
      <div class="input-box">
        <span class="icon"><ion-icon name="image"></ion-icon></span>
        <input type="file" name="profile_pic">
        <label>Gambar Profil</label>
      </div>
      <div class="remember-forgot">
        <label><input type="checkbox"> Saya menyetujui syarat dan ketentuan</label>
      </div>
      <button type="submit" class="btn">Register</button>
    </form>
  </div>
</div>

```

- Kode Halaman Index

Kode ini digunakan untuk menampilkan halaman *index* yang berisi artikel terkait 10 jenis penyakit kulit.

```

<div class="container mt-5">
  <div class="row">
    <div class="col-md-6 col-sm-6">
      <div class="card mb-4">
        <a href="detail_jerawat"></a>
        <div class="card-body">
          <h5 class="card-title">Jerawat</h5>
          <p class="card-text">Kenali Jenis, Ciri, Penyebab, dan Cara Pencegahan Jerawat</p>
        </div>
      </div>
    </div>
  </div>
</div>

```

- Kode Halaman Deteksi

Kode ini digunakan untuk menampilkan halaman deteksi yang memuat tombol "*Realtime Capture*" dan "*Upload Image*", yang memungkinkan pengguna untuk memilih metode deteksi yang ingin digunakan.

```
<div class="container mt-5">
  <div class="row align-items-center">
    <div class="col-lg-6 col-md-6 col-sm-12 order-lg-2 order-md-2 order-sm-1">
      <div class="gambar">
        
      </div>
    </div>
    <div class="col-lg-6 col-md-6 col-sm-12 order-lg-1 order-md-1 order-sm-2">
      <div class="text">
        <h2 class="deteksi fs-2">Deteksi kondisi kulitmu di sini.</h2>
        <p class="deteksi-desc fs-5">Temukan rekomendasi obat yang tepat dan pahami kondisi kulitmu dengan mudah. Aplikasi ini dapat
        <div class="d-flex justify-content-start">
          <button type="button" class="btn btn-primary deteksi-button mt-3 me-3" onclick="startDetection()">Deteksi</button>
          <button type="button" class="btn btn-primary deteksi-button mt-3" onclick="goToUploadPage()">Upload</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

- Kode Halaman Realtime Capture

Kode ini digunakan untuk menampilkan halaman *realtime capture*, di mana pengguna dapat melakukan deteksi secara langsung menggunakan kamera perangkat yang digunakan.

```
<body>
  <h1>Deteksi dan Klasifikasi Penyakit Kulit</h1>
  <div>
    
  </div>
  <button id="capture-button" onclick="captureImage()">Capture</button>
  <div id="result">Menunggu hasil klasifikasi...</div>
</body>
```

- Kode Halaman Upload Image

Kode ini digunakan untuk menampilkan halaman *upload image*, di mana pengguna dapat melakukan deteksi penyakit kulit dengan mengunggah foto yang mungkin menunjukkan adanya penyakit kulit.

```
<main>
  <div class="upload-container">
    <h1>Upload Gambar</h1>
    <p class="upload-description">Pastikan gambar yang Anda pilih memiliki kualitas jelas dan sesuai dengan kebutuhan
    <form id="upload-form" action="/upload" method="post" enctype="multipart/form-data">
      <input type="file" name="file" id="file" accept="image/*">
      <input type="submit" value="Upload">
    </form>
    <div id="result"></div>
  </div>
</main>
```

- Kode Halaman Chatbot

Kode ini digunakan untuk menampilkan halaman *chatbot*

```

<div class="container mt-5">
  <section class="msger">
    <main class="msger-chat">
      <div class="msg left-msg">
        
        <div class="msg-bubble">
          <div class="msg-info">
            <div class="msg-info-name">Chatbot</div>
            <div class="msg-info-time">{{ current_time }}</div>
          </div>
          <div class="msg-text">
            Hai, selamat datang di ChatBot! Silakan kirim saya pesan.
          </div>
        </div>
      </div>
    </div>
  </div>

```

- Kode Halaman Profil

Kode ini digunakan untuk menampilkan halaman profil pengguna. Di halaman ini, pengguna dapat melihat dan mengedit informasi pribadi, seperti nama, *username*, dan email yang telah didaftarkan saat *registrasi*. Pengguna dapat menggunakan halaman ini untuk mengubah data profil serta mengganti gambar profil sesuai keinginan.

```

<div class="container profile-section">
  <div class="row">
    <div class="col-md-8 offset-md-4">
      <div class="row align-items-center">
        <div class="col-md-6 text-center">
          
        </div>
        <div class="col-md-6 profile-details">
          <h2>Profil</h2>
          <form id="profileForm" enctype="multipart/form-data">
            <div class="form-group">
              <label for="fullName">Nama</label>
              <input type="text" id="fullName" name="fullName" value="{{ user['fullname'] }}"
                class="form-control">
            </div>
            <div class="form-group">
              <label for="username">Username</label>
              <input type="text" id="username" name="username" value="{{ user['username'] }}"
                class="form-control">
            </div>
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" id="email" name="email" value="{{ user['email'] }}"
                class="form-control">
            </div>
            <div class="form-group file-upload">
              <label for="profilePic">Foto</label>
              <input type="file" id="profilePic" name="profilePic" class="form-control">
            </div>
            <div class="form-group">
              <a href="{{ url_for('change_password') }}" class="btn btn-danger">Ubah Password</a>
              <button type="submit" class="btn btn-success">Simpan</button>
              <button type="button" class="btn btn-secondary">Batal</button>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>

```

- Kode Halaman Ubah Password

Kode ini digunakan untuk menampilkan halaman ubah *password*, jika pengguna ingin melakukan perubahan dengan memasukkan *password* lama, *password* baru, dan mengkonfirmasi *password* baru.

```
<div class="container profile-section">
  <div class="row">
    <div class="col-md-12">
      <h2>Ubah Password</h2>
      <form action="{{ url_for('change_password') }}" method="POST">
        <div class="form-group">
          <label for="current_password">Password Lama</label>
          <input type="password" id="current_password" name="current_password" class="form-control" required>
        </div>
        <div class="form-group">
          <label for="new_password">Password Baru</label>
          <input type="password" id="new_password" name="new_password" class="form-control" required>
        </div>
        <div class="form-group">
          <label for="confirm_new_password">Konfirmasi Password</label>
          <input type="password" id="confirm_new_password" name="confirm_new_password" class="form-control" required>
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-success">Ubah Password</button>
          <a href="{{ url_for('profile') }}" class="btn btn-secondary">Cancel</a>
        </div>
      </form>
    </div>
  </div>
</div>
```

Lampiran 5 Sertifikat HKI Yang Telah Terbit


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 : EC00202466227, 16 Juli 2024

Pencipta

Nama : Naovi Magfiroh, Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng. dkk

Alamat : Jln. DR. Sarjito No. 28, RT 05 RW 03, Kelurahan Gandasuli Kec.Brebes Kab.Brebes Provinsi Jawa Tengah 52215, Brebes, Brebes, Jawa Tengah, 52215

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

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

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

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Aplikasi Deteksi Penyakit Kulit Menggunakan Arsitektur Resnet50 Dengan Memanfaatkan Teknologi Chatbot Berbasis Website

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 16 Juli 2024, di Tegal

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

Nomor pencatatan : 000641582

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	Naovi Magfiroh	Jln. DR. Sarjito No. 28, RT 05 RW 03, Kelurahan Gandasuli Kec.Brebes Kab.Brebes Provinsi Jawa Tengah 52215, Brebes, Brebes
2	Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.	Jln. Raya Kluwut Timur No. 24, Kelurahan Kluwut Kec.Bulakamba Kab.Brebes Provinsi Jawa Tengah 52253, Bulakamba, Brebes
3	Hepatika Zidny Ilmadina, S.Pd., M.Kom.	Jln. Kenanga Gang 1 No. 9, Kelurahan Mangkukusan Kecamatan Tegal Timur Kota Tegal Provinsi Jawa Tengah 52123, Tegal Timur, Tegal



Lampiran 6 Lembar Bimbingan



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Naovi Magfiroh
 Nim : 20090135
 No. Ponsel : 083844823428
 Judul TA : Aplikasi Deteksi Penyakit Kulit Menggunakan Algoritma
Convolutional Neural Network (CNN) Dengan
 Memanfaatkan Teknologi *Chatbot Berbasis Website*.
 Dosen Pembimbing I : Ir. Ginanjar Wiro Sasmito, M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1	21/03/24	Konsep UI/UX	Ok.	f.
2.	28/03/24	UI/UX	→ lanjut produk agar bisa jalan.	f.
3.	4/04/2024	UI/UX	Ok. Lanjut ke Aplikasi	f.
4.	9/06/2024	produk	- Validasi produk. - chatbot.	f.
5	12/06/2024	produk	Chatbot Ok. → Depresi Real time.	f.



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
6	1/7/21	proklam	ole.	f.
7	17/7/21	Laporan	Revizi	f.
8	18/7/21	Laporan	ole. himp usj	f.



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing

Tegal, 10 Juni 2024
Dosen Pembimbing I

Ir. Gnanjar Wiro Sasmito, M.Kom.
NIPY. 10.007.032



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Naovi Magfiroh
Nim : 20090135
No. Ponsel : 083844823428
Judul TA : Aplikasi Deteksi Penyakit Kulit Menggunakan Algoritma
Convolutional Neural Network (CNN) Dengan
Memanfaatkan Teknologi Chatbot Berbasis Website.
Dosen Pembimbing II : Hepatika Zidny Ilmadina, S.Pd., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	22/03/2023	* metode * alur aplikasi	* tetapkan metode yg akan digunakan dalam deteksi * tambah beberapa kelas seperti birang kerungat, meloma, dll * deteksi realtime ↓ auto capture (misal 2 detik) ↓ mulai kelas + rekomendasi	
2.	23/04/2023	* metode	* deteksi → cascade aja → buat sendiri di MF PCD ada * klasifikasi ke map kelas pake PerNetso ↓ cari teknik fine tuning - sementara pake learning from scratch hasil overfitting * kalau udah nemu fine tuning, save model → kombinasi pake cascade	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
3.	15/05/2024	*metode	<p>x to do list:</p> <ol style="list-style-type: none"> 1) perbaiki deteksi haar cascade supaya hanya capture penyakit saja 2) usikan dg model yg mudah ditraining 3) perbaiki training modelnya lagi 	
4.	29/05/2024	*metode	<p>to do list:</p> <ol style="list-style-type: none"> 1) sinkron ke frontend 2) perbaiki model <ul style="list-style-type: none"> - fine tuning - hypermater model 	
5.	03/06/2024	*pradik	<p>to do list:</p> <ol style="list-style-type: none"> 1) memisahkan dari to kelas yang membutuhkan dokter / tidak <ul style="list-style-type: none"> ↓ "Hubungi / kunjungi Fasilitas terdekat utk mendapatkan pemeriksaan lanjutan" ⇒ gambar obat hanya 7meringankan gejala ⇒ tdk boleh yg antibiotik 2) periksa kembali y/ code prediksi ⇒ brskali salah akurasi cuma 8% 	
6.	04/07/2024	*produk	<p>to do list:</p> <ol style="list-style-type: none"> 1. manual book 1. berkas HFI 3. technical report 	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
7.	11/7/2024	* cekas HKI	* mendaftar HKI * menyiapkan technical report penelitiannya	
8.	17/7/2024	* laporan	* revisi sesuai catatan	
9.	18/7/2024	* laporan	* acc laporan * siapkan berkas sidang	

Tegal, 18 Juli 2024
Dosen Pembimbing II

Hepatika Zidny Ilimadina, S.Pd., M.Kom.
NIPY. 08.017.340