LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama	: Lulu Nadhiatun Anisa
NIM	: 21090066
Program Studi	: Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama	: Dwi Intan Af'idah, S.T., M.Kom.
Status	: Dosen
NIDN	: 0620089203
Jabatan Fungsional	: Lektor
Pangkat/Golongan	: Penata III/C

Pada hari ini Kamis tanggal 06 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan dengan syarat adanya peningkatan progres dari proyek skripsi yang dilaporkan setiap satu pekan. Jika Pihak 1 tidak memenuhi syarat tersebut, maka Pihak 2 berhak memutuskan kesepakatan ini.
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 06 Maret 2025

Pihak Pertama	Pihak Kedua
Lulu Nadhiatun Anisa	Dwi Intan Af'idah, S.T., M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika

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

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama	: Lulu Nadhiatun Anisa
NIM	: 21090066
Program Studi	: Sarjana Terapan 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 Rabu tanggal 05 Maret 2025 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, 05 Maret 2025

Pihak Pertama	Pihak Kedua
Lulu Nadhiatun Anisa	Hepatika Zidny Ilmadina, S.Pd., M.Kom.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika

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

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Lulu Nadhiatun Anisa
   Kewarganegaraan : Indonesia
   Alamat : Desa Sumurpanggang RT.01/ RW.01, Kecamatan Margadana, Kota Tegal

2. Nama : Dwi Intan Af'idah
   Kewarganegaraan : Indonesia
   Alamat : Desa Grinting RT.03/ RW.02, Kecamatan Bulakamba, Kabupaten Brebes

3. Nama : Hepatika Zidny Ilmadina
   Kewarganegaraan : Indonesia
   Alamat : Jalan Kenanga Gang I/9, RT. 03/ RW. 01, Kelurahan Mangkukusuman, Kecamatan Tegal Timur, Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
   Berupa : Program Komputer
   Berjudul : Aplikasi Deteksi Dini Anemia Melalui Konjugtiva Mata Untuk Perempuan Usia Produktif Sebagai Upaya Pencegahan Stunting

   - 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 sebagimana mestinya.

Tegal, 19 Juni 2025

(Lulu Nadhiatun Anisa)
Pemegang Hak Cipta*

(Dwi Intan Af'idah)
Pemegang Hak Cipta*

(Hepatika Zidny Ilmadina)
Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Lulu Nadhiatun Anisa
   Kewarganegaraan : Indonesia
   Alamat : Desa Sumurpanggang RT.01/ RW.01, Kecamatan Margadana, Kota Tegal

2. Nama : Dwi Intan Af'idah
   Kewarganegaraan : Indonesia
   Alamat : Desa Grinting RT.03/ RW.02, Kecamatan Bulakamba, Kabupaten Brebes

3. Nama : Hepatika Zidny Ilmadina
   Kewarganegaraan : Indonesia
   Alamat : Jalan Kenanga Gang I/9, RT. 03/ RW. 01, Kelurahan Mangkukusuman, 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 : Jalan Mataram Nomor 9, Kelurahan Pesurungan Lor,
Kecamatan Margadana, Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Deteksi Dini Anemia Melalui Konjugtiva Mata Untuk Perempuan Usia Produktif Sebagai Upaya Pencegahan Stunting" 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, 19 Juni 2025

Pemegang Hak Cipta
Kepala P3M

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

Pencipta

(Lulu Nadhiatun Anisa)

(Dwi Intan Af'idah)

(Hepatika Zidny Ilmadina)

Lampiran 4. Manual Book dan Dokumen Teknikal

Latar Belakang

Stunting adalah kondisi kekurangan gizi kronis di masa kehamilan yang berdampak pada gangguan perkembangan otak, kemampuan kognitif dalam janin dan pertumbuhan fisik yang lebih pendek dari standar usianya. Faktor utama stunting adalah gizi buruk yang dialami oleh balita selama masa kehamilan dan anak selama masa balita. Pengobatan stunting sebaiknya tidak hanya dimulai ketika balita sudah mengalami stunting, tetapi perlu dimulai sejak calon ibu merencanakan kehamilannya.

Ibu hamil yang mengalami anemia memiliki risiko empat kali lebih tinggi melahirkan anak stunting dibandingkan ibu hamil yang sehat. Perempuan usia produktif yang mengalami anemia saat menstruasi juga berisiko lebih tinggi terkena anemia saat hamil. Anemia merupakan penyakit ketiga yang paling sering dialami ibu hamil dan menjadi salah satu penyebab stunting. Jika tidak ditangani, anemia pada ibu hamil dapat menghambat pertumbuhan janin dan meningkatkan risiko komplikasi, seperti kelahiran prematur, bayi dengan berat lahir rendah, hingga kematian ibu dan bayi. Beberapa faktor penyebab tingginya kasus anemia pada ibu hamil antara lain pernikahan dini dan kehamilan yang tidak direncanakan. Kurangnya akses layanan kesehatan di daerah pedesaan juga memperburuk situasi ini dan perlu mendapat perhatian serius.

Pencegahan untuk peningkatan kasus anemia pada ibu hamil, diperlukan sistem deteksi dini yang efektif agar lebih hemat waktu, biaya, dan mudah diakses tanpa kendala jarak. Berdasarkan permasalahan yang telah ada, pembuatan aplikasi EyeStunt menjadi langkah penting dalam membantu perempuan usia produktif dengan mendeteksi anemia secara dini sebagai upaya pencegahan stunting melalui pemanfaatan teknologi kecerdasan buatan. Aplikasi ini menyediakan solusi yang cepat, akurat, dan terjangkau untuk mendeteksi anemia sejak dini secara non-invasif, sehingga prosesnya lebih mudah dan efektif.

Panduan Pengguna

Tampilan Awal

Pengguna dapat melakukan instal aplikasi EyeStunt terlebih dahulu. Setelah proses instalasi selesai, pengguna dapat langsung membuka aplikasi

Splash Screen

Saat pertama kali membuka aplikasi EyeStunt, pengguna akan disambut oleh splash screen yang menampilkan logo dan identitas aplikasi sebagai tampilan awal sebelum masuk ke halaman utama.

Welcome Activity

Setelah splash screen, pengguna akan diarahkan ke halaman welcome activity sebagai pengantar sebelum melakukan login atau registrasi.

Pengguna dapat memilih untuk melakukan registrasi jika belum memiliki akun atau login jika sudah memiliki akun dan dapat mengakses aplikasi EyeStunt

Registrasi, Assesment & Login

Pastikan bahwa pengguna sudah terhubung dengan jaringan internet untuk mengakses aplikasi EyeStunt.

Step 1

Pengguna dapat mendaftar dengan mengisikan nama lengkap, email, tanggal lahir dan password. Registrasi hanya bisa dilakukan bagi pengguna berusia 18–40 tahun. Namun, Jika pengguna sudah melakukan registrasi sebelumnya, maka pengguna dapat langsung login ke aplikasi EyeStunt.

Step 2

Setelah melakukan registrasi, pengguna akan diarahkan untuk menjawab empat pertanyaan asesment awal. Asesment ini berfungsi untuk mengidentifikasi kategori pengguna, yaitu apakah termasuk ibu hamil, perencana kehamilan, atau pengguna biasa (remaja dewasa putri). Setiap asesment dapat diisi oleh pengguna sesuai dengan kondisi masing-masing. Setelah seluruh asesment diisi, pengguna dapat menekan tombol 'Selesai', kemudian melanjutkan dengan login ke aplikasi EyeStunt.

Step 3

Pengguna dapat melakukan login ke aplikasi jika sudah melakukan registrasi dan pengisian assesment. Pengguna dapat memasukkan email dan password yang telah dibuat sebelumnya.

Beranda Pengguna

Setelah pengguna berhasil membuat akun dan selesai mengisikan asesment awal, pengguna akan mendapatkan akses penuh aplikasi EyeStunt dengan masuk ke halaman beranda.

Profile Pengguna
Pengguna dapat melihat profile akun serta informasi lengkap yang telah terdaftar di aplikasi EyeStunt.

Deteksi Anemia
Digunakan untuk mendeteksi anemia pada perempuan usia produktif dengan konjugtiva mata.

History
Menampilkan hasil riwayat deteksi yang dilakukan pengguna sebelumya untuk mengetahui anemia atau normal secara berkala.

Fasilitas Kesehatan Terdekat
Fitur ini, hanya bisa diakses oleh ibu hamil yang akan memberikan rekomendasi puskesman, klinik dan rumah sakit terdekat dari pengguna

Konsultasi Dokter
Digunakan untuk berkonsultasi dengan dokter apabila pengguna telah melakukan deteksi dan memperoleh hasil dengan akurasi lebih dari 85%.

Artikel Kesehatan
Menampilkan artiker yang mencakup informasi serta pengetahuan mengenai anemia dan stunting.

Chatbot
Dapat memberikan respon 24/7 dan real time. Serta dapat memberikan pertanyaan seputar anemia.

Deteksi Anemia

Dalam mengakses fitur ini, Pengguna diberikan dua opsi untuk memulai proses deteksi, yaitu melalui kamera realtime atau mengunggah foto.

1. Deteksi anemia dengan kamera

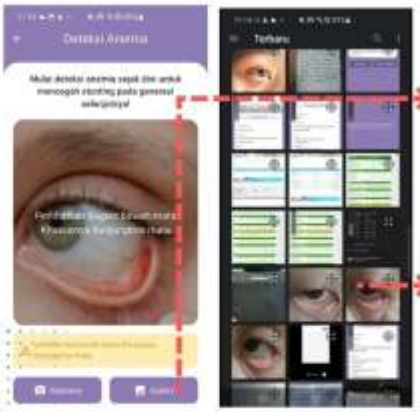
Step 1
Jika pengguna ingin mengambil gambar secara langsung, dapat mengklik tombol "Kamera".

Step 2
Jika pengguna memilih menggunakan kamera, pengguna dapat masuk ke halaman kamera. Pada halaman ini, pengguna diminta untuk memposisikan konjungtiva mata sesuai dengan icon yang ditampilkan di layar.

Step 3

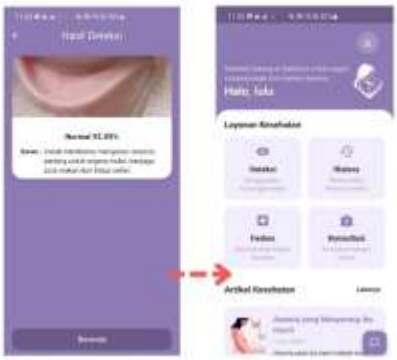
- Terdapat fitur zoom in dan zoom out untuk membantu pengguna mendapatkan tampilan yang lebih jelas dan sesuai.
- Selain itu terdapat menu switch untuk pengambilan gambar menggunakan kamera belakang.
- Setelah posisi dirasa tepat, pengguna dapat menekan tombol kamera untuk mengambil gambar.

2. Deteksi anemia dengan unggah galeri



Step 1
Jika pengguna sudah memiliki gambar konjungtiva mata yang tersimpan dapat memilih tombol "Galeri".

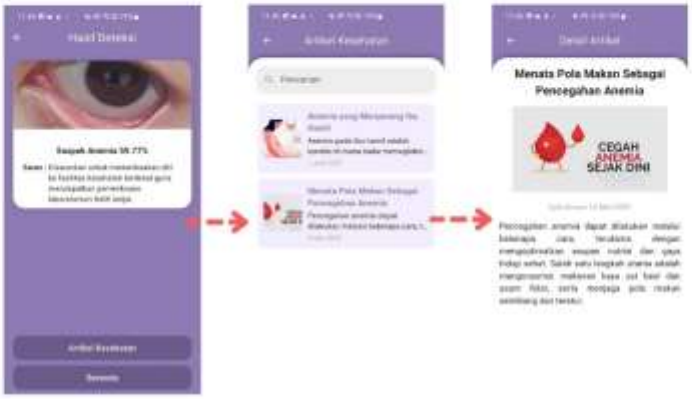
Step 2
Pengguna dapat memilih gambar konjungtiva mata yang sudah tersedia di perangkat. Disarankan untuk memilih gambar yang memperlihatkan konjungtiva mata dengan jelas dan dalam jarak dekat (close-up), sehingga area konjungtiva tampak detail dan tidak buram.



Setelah proses deteksi dilakukan, pengguna diminta untuk menunggu beberapa saat agar sistem aplikasi EyeStunt melakukan analisis gambar konjungtiva yang telah diunggah. Jika pengguna terdeteksi normal hanya akan diarakhan ke halaman beranda. Hasil deteksi yang ditampilkan dapat berbeda, tergantung pada kategori pengguna seperti ibu hamil, perencana kehamilan, atau remaja dewasa putri.

Selain itu, untuk hasil deteksi yang ditampilkan dapat berbeda itu ketika pengguna teridentifikasi "Suspek Anemia" pada kategori pengguna seperti ibu hamil, perencana kehamilan, atau remaja dewasa putri.

a. Hasil Deteksi Suspek Anemia pada remaja dewasa muda

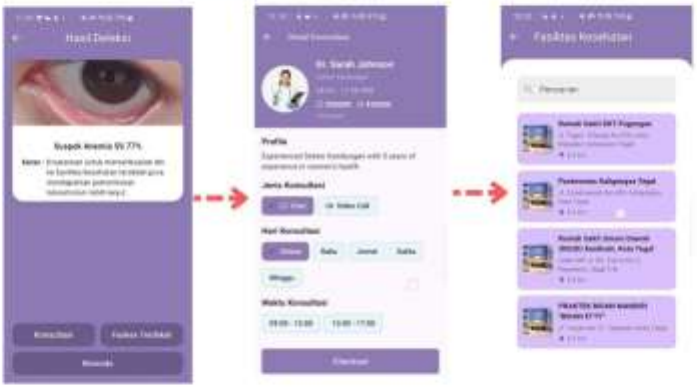


Ketika remaja dewasa putri melakukan pendeteksian dan teridentifikasi "Suspek Anemia, sistem akan menampilkan hasil berupa akurasi, analisis dan saran yang perlu dilakukan. Selain itu, pengguna juga akan diberikan akses ke menu artikel kesehatan yang berisi informasi mengenai cara menjaga pola makan, mencegah anemia, pencegahan stunting dan tips kesehatan lainnya. Setelah melihat hasil deteksi, pengguna dapat kembali ke halaman utama aplikasi melalui menu Beranda.

b. Hasil Deteksi Suspek Anemia pada perencana kehamilan

Jika pengguna yang merencanakan kehamilan terdeteksi sebagai "Suspek Anemia", sistem akan menampilkan hasil berupa akurasi, analisis, dan saran penanganan. Selanjutnya, pengguna dapat mengakses menu Konsultasi Dokter untuk berkonsultasi langsung dengan tenaga medis. Dalam menu ini, pengguna bisa melihat profil dokter, jadwal ketersediaan, biaya konsultasi, dan melakukan pemesanan setelah pembayaran. Setelah itu, pengguna dapat kembali ke halaman utama melalui menu Beranda.

b. Hasil Deteksi Suspek Anemia pada ibu hamil



Jika pengguna yang sedang hamil terdeteksi sebagai "Suspek Anemia", sistem akan menampilkan hasil berupa akurasi, analisis kondisi, dan saran. Sebagai tindak lanjut, pengguna dapat mengakses dua menu: Konsultasi untuk berkonsultasi online dengan dokter, dan Faskes Terdekat untuk menemukan lokasi pelayanan kesehatan di sekitar mereka. Pada menu Konsultasi, pengguna bisa melakukan booking pemeriksaan, sedangkan menu Faskes menampilkan rekomendasi lokasi beserta informasi detail. Setelah itu, pengguna dapat kembali ke halaman utama melalui menu Beranda.

Konsultasi Dokter

Step 1

Pengguna yang terindikasi suspek anemia (baik ibu hamil maupun yang sedang merencanakan kehamilan), ketika mengklik tombol 'Konsultasi', akan diarahkan ke daftar dokter yang dapat membantu mereka melakukan konsultasi melalui chat online.

Step 2

Setelah memilih dokter, pengguna akan diarahkan ke halaman detail dokter. Sistem menyediakan dua opsi konsultasi, yaitu melalui chat atau video call, yang dapat dipilih sesuai kebutuhan.

Step 3

Sistem menampilkan jadwal konsultasi yang tersedia untuk dipilih oleh pengguna.

Step 4

Pengguna dapat memilih waktu konsultasi dengan dokter berdasarkan sesi yang masih tersedia.

Step 5

Setelah memilih jenis konsultasi, jadwal, dan waktu yang tersedia, pengguna dapat melanjutkan ke proses checkout.

Step 5

Setelah menyelesaikan checkout untuk sesi konsultasi, sistem akan menampilkan halaman detail pembayaran. Pengguna dapat memilih metode pembayaran yang ingin digunakan.

Step 6

Setelah memilih metode pembayaran, pengguna dapat menyelesaikan pembayaran sesuai dengan nominal yang tertera pada total pembayaran. Selanjutnya, sistem akan memproses pembayaran untuk sesi konsultasi tersebut.

Step 7

Setelah berhasil memesan sesi konsultasi, pengguna dapat mengikuti sesi konsultasi sesuai dengan jadwal yang telah dipilih.

Advice Dokter

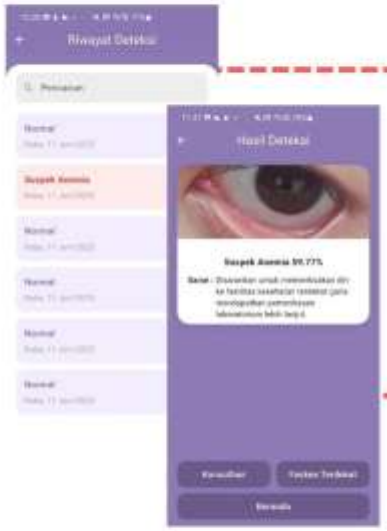
Step 1

Setelah menyelesaikan sesi konsultasi dengan dokter, tombol "Lihat Advice" akan muncul. Pengguna dapat mengakses halaman Advice untuk melihat dan menerima saran atau rekomendasi yang diberikan oleh dokter.

Step 2

Setelah dapat mengakses detail dari advice, pengguna juga dapat melakukan download hasil advice ke pdf.

Riwayat Deteksi



Step 1

Fitur ini dapat dilihat jika pengguna telah melakukan deteksi sebelumnya. Jika belum, riwayat deteksi akan ditampilkan "Belum ada riwayat". Pengguna dapat melihat detail hasil deteksi yang telah dilakukan dengan memilih salah satu riwayat yang tersedia.

Step 2

Riwayat hasil deteksi sebelumnya dapat diakses melalui fitur History. Jika pengguna membuka hasil deteksi yang menunjukkan suspek anemia, maka sistem akan menyediakan akses bebrapa akses ke fitur konsultasi atau Fasilitas kesehatan terdekat sesuai dengan sataus penggunanya.

Artikel Kesehatan

Fitur ini menyediakan berbagai artikel informatif seputar anemia dan stunting pada anak. Melalui fitur ini, pengguna dapat mencari dan mempelajari informasi penting untuk menambah pengetahuan mengenai kedua kondisi tersebut.

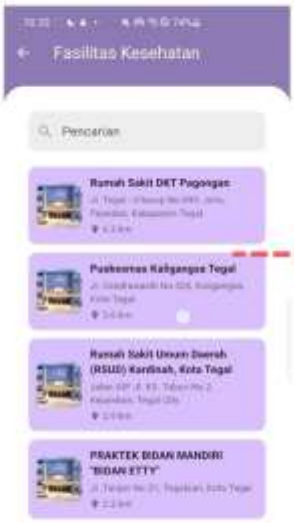
Step 1

Pengguna dapat mengakses daftar artikel kesehatan dengan menekan menu 'Lainnya', yang akan mengarahkan ke halaman berisi seluruh artikel yang tersedia di aplikasi. pengguna juga dapat melakukan pencarian artikel pada fitur tersebut. Selain itu, pengguna juga dapat langsung membuka artikel dengan memilih salah satu artikel yang ditampilkan di beranda.

Step 2

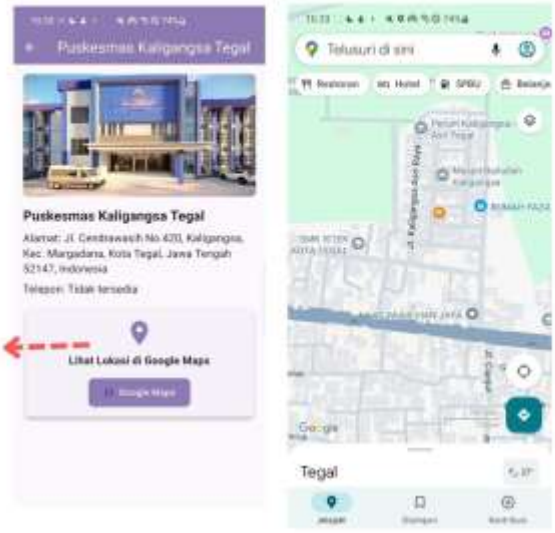
Setelah memilih salah satu card artikel, pengguna akan diarahkan ke halaman detail yang menampilkan isi lengkap artikel tersebut.

Fasilitas Kesehatan Terdekat



Step 1

Fitur ini hanya bisa diakses ketika pengguna, dalam masa kehamilan dan terdeteksi suspek anemia. Fitur ini dapat menampilkan list rekomendasi pusekesmas, klinik, dan rumah sakit terdekat dari posisi pengguna mengakes apikasi EyeStunt untuk penangan anemia pada masa kehamilan yang lebih baik.



Step 2

Pengguna dapat melihat detail rumah terdekat yang dipilih untuk melakukan konsultasi langsung dengan dokter guna penanganan anemia selama kehamilan.

Step 3

Jika pengguna mengklik tombol "Google Maps" pengguna dapat mengakses lokasi fasilitas kesehatan yang telah dipilih melalui integrasi dengan Google Maps.

Profile Pengguna

Pengguna dapat mengakses bagian profile dengan mengklik menu profil, dimana pengguna dapat melihat informasi lengkap terkait akun yang telah terdaftar, seperti nama, email, dan data pribadi lainnya.

1. Mengubah Status Kehamilan

Status kehamilan dapat diubah dengan menekan tombol 'Hamil'. Tombol ini berfungsi sebagai saklar on/off: warna abu-abu menandakan nonaktif, sedangkan hijau menandakan aktif.

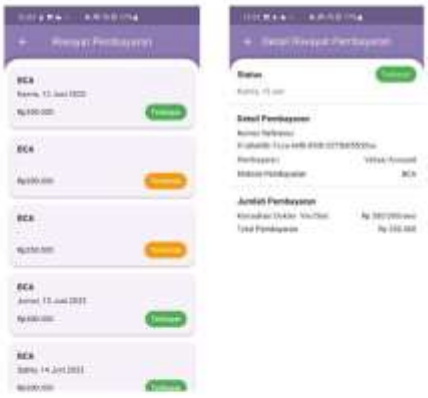
2. Edit Profile Pengguna

Pengguna dapat melakukan update profile dengan mengklik "Edit Profile". Pengguna dapat mengedit nama lengkap, email, dan tanggal lahir. Setelah melakukan perubahan, pengguna dapat menekan tombol 'Update Profile' untuk menyimpan pembaruan.

3. Ubah Password

Pengguna dapat melakukan update password dengan mengklik "Ubah Password". Pengguna dapat menginputkan password sebelumnya, password baru dan konfirmasi password ulang. Setelah melakukan perubahan, pengguna dapat menekan tombol 'Update Password' untuk menyimpan pembaruan.

4. Riwayat Pembayaran



Step 1

Pengguna dapat mengakses riwayat pembayaran di aplikasi. Jika pengguna pernah melakukan transaksi, maka akan muncul daftar riwayat transaksi untuk sesi konsultasi yang pernah dilakukan.

Step 2

Pengguna dapat menakses detail dari riwayat pembayaran yang telah dilakukan

5. Tentang Aplikasi

Pengguna juga dapat mengakses tentang aplikasi yang berisikan informasi tentang aplikasi EyeStunt dengan mengklik "Tentang Aplikasi".

6. Keluar

Pengguna dapat melakukan logout akun dengan mengkik tombol "Keluar".

Chatbot

Pengguna dapat mengakses fitur Chatbot bernama EyeStuntBot yang siap merespons pertanyaan secara real-time selama 24 jam setiap hari. Melalui EyeStuntBot, pengguna dapat mengajukan berbagai pertanyaan seputar anemia dan stunting, dan akan mendapatkan jawaban yang cepat dan informatif.

Pengguna dapat mengirimkan pertanyaan seputar anmia dan stunting yang akan di respon secara real-time oleh EyeStuntBot.

Panduan **Dokter**

Sama seperti dengan Pengguna, Dokter dapat melakukan login ke aplikasi jika sudah memiliki akun dan hak akses aplikasi EyeStunt. Dokter dapat melakukan login dengan memasukkan email dan password yang telah dibuat oleh admin.

Beranda

Setelah berhasil login ke aplikasi EyeStunt, dokter dapat mengakses berbagai fitur yang tersedia di dalam aplikasi.

Jadwal Konsultasi
Dokter dapat mengatur jadwal konsultasi dengan pasien.

Profile Dokter
Dokter dapat melihat informasi lengkap Dokter

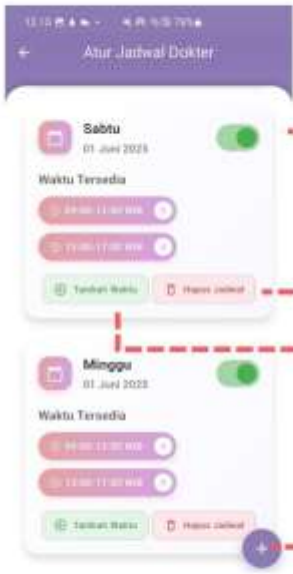
Pasien
Dokter dapat melilhat list pasien yang akan melakukan konsultasi sesuai tanggal sesi yang akan dilakukan

Konsultasi
Dokter dapat melihat riwayat konsultasi dan jumlah konsultasi yang sudah dilakukan dengan pasien .

Riwayat
Dokter dapat melihat riwayat konsultasi yang pernah dilakukan dari pasien.

Statistik
Menampilkan sttsistik data untuk dokter.

Jadwal Konsultasi



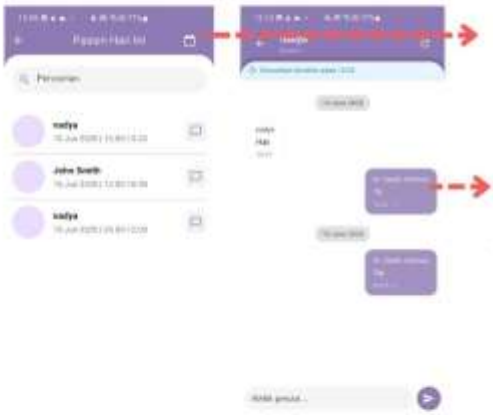
Dokter dapat menonaktifkan jadwal yang sudah dibuat.

Dokter dapat menghapus jadwal konsultasi yang sudah di buat.

Dokter dapat menambahkan hari dan jam sesuai dengan sesi yang akan dilakukan

Dokter dapat melakukan tambah hari dengan menyesuaikan jadwal yang diinginkan.

Pasien



Dokter dapat melakukan filtering data pasien untuk berkonsultasi berdasarkan tanggal

Dokter dapat melakukan chat dengan pasien sesuai dengan jam dan hari dari sesi yang ada

Advice Dokter

Dokter dapat melakukan tambah advice ke pasien ketika sesi konsultasi telah berakhir dan dapat melakukan edit advice yang telah dibuat.

Riwayat

Dokter dapat melihat daftar pasien yang sudah melakukan konsultasi sesuia dengan tanggal dan hari.

Konsultasi

Dokter dapat melihat daftar pasien yang sudah melakukan konsultasi sesuia dengan tanggal dan hari.

16

Profile

Dokter dapat mengakses bagian profile dengan mengklik menu profile, dimana dokter dapat melihat informasi lengkap terkait akun yang telah terdaftar, seperti nama, email, dan tanggal lahir.

1. Edit Profile Pengguna

Dokter dapat melakukan update profile dengan mengklik "Edit Profile". Pengguna dapat mengedit nama lengkap, email, dan tanggal lahir. Setelah melakukan perubahan, pengguna dapat menekan tombol 'Update Profile' untuk menyimpan pembaruan.

2. Ubah Password

Pengguna dapat melakukan update password dengan mengklik "Ubah Password". Pengguna dapat menginputkan password sebelumnya, password baru dan konfirmasi password ulang. Setelah melakukan perubahan, pengguna dapat menekan tombol 'Update Password' untuk menyimpan pembaruan.

3. Tentang Aplikasi

Pengguna juga dapat mengakses tentang aplikasi yang berisikan informasi tentang aplikasi EyeStunt dengan mengklik "Tentang Aplikasi".

4. Keluar

Pengguna dapat melakukan logout akun dengan mengkik tombol "Keluar".

Dokumen Teknis

EYESTUNT

Deteksi Dini Anemia, Cegah Stunting
Sejak Perencanaan Kehamilan!

Penulis :
Lulu Nadhiatun Anisa
Dwi Intan Af'idah, S.T., M.Kom.
Hepatika Zidnyilmadina, S.Pd., M.Kom

Latar **Belakang**

Stunting adalah kondisi kekurangan gizi kronis di masa kehamilan yang berdampak pada gangguan perkembangan otak, kemampuan kognitif dalam janin dan pertumbuhan fisik yang lebih pendek dari standar usianya. Faktor utama stunting adalah gizi buruk yang dialami oleh balita selama masa kehamilan dan anak selama masa balita. Pengobatan stunting sebaiknya tidak hanya dimulai ketika balita sudah mengalami stunting, tetapi perlu dimulai sejak calon ibu merencanakan kehamilannya.

Ibu hamil yang mengalami anemia memiliki risiko empat kali lebih tinggi melahirkan anak stunting dibandingkan ibu hamil yang sehat. Perempuan usia produktif yang mengalami anemia saat menstruasi juga berisiko lebih tinggi terkena anemia saat hamil. Anemia merupakan penyakit ketiga yang paling sering dialami ibu hamil dan menjadi salah satu penyebab stunting. Jika tidak ditangani, anemia pada ibu hamil dapat menghambat pertumbuhan janin dan meningkatkan risiko komplikasi, seperti kelahiran prematur, bayi dengan berat lahir rendah, hingga kematian ibu dan bayi. Beberapa faktor penyebab tingginya kasus anemia pada ibu hamil antara lain pernikahan dini dan kehamilan yang tidak direncanakan. Kurangnya akses layanan kesehatan di daerah pedesaan juga memperburuk situasi ini dan perlu mendapat perhatian serius.

Pencegahan untuk peningkatan kasus anemia pada ibu hamil, diperlukan sistem deteksi dini yang efektif agar lebih hemat waktu, biaya, dan mudah diakses tanpa kendala jarak. Berdasarkan permasalahan yang telah ada, pembuatan aplikasi EyeStunt menjadi langkah penting dalam membantu perempuan usia produktif dengan mendeteksi anemia secara dini sebagai upaya pencegahan stunting melalui pemanfaatan teknologi kecerdasan buatan. Aplikasi ini menyediakan solusi yang cepat, akurat, dan terjangkau untuk mendeteksi anemia sejak dini secara non-invasif, sehingga prosesnya lebih mudah dan efektif.

Source **Code**

1. Import Library

```python
import traceback
from flask import Blueprint, logging, request, jsonify, render_template,redirect, url_for, flash, session
from flask_jwt_extended import create_access_token, get_jwt_identity, jwt_required
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime, timedelta
from werkzeug.utils import secure_filename
from app.models import HealthArticle, Payment, User, UserRole
from app import db
from functools import wraps
from sqlalchemy import or_
import os
```

```python
from app import create_app, db
from app.models import (
    User, UserRole, Questionnaire, HistoryDetection,
    DoctorProfile, DoctorSpecialty, DoctorSchedule, DayOfWeek, DoctorTimeSlot,
    Consultation, ConsultationStatus,
    Payment, PaymentStatus, PaymentMethod, PaymentProvider
)
from datetime import datetime, date, time, timedelta
from werkzeug.security import generate_password_hash
import random
import uuid
```

Bagian ini merupakan import library yang digunakan untuk mendukung fungsionalitas aplikasi berbasis Flask. Library flask dan flask_jwt_extended merupakan inti dari aplikasi, di mana Flask menyediakan kerangka kerja dasar untuk web development sementara Flask-JWT-Extended menangani autentikasi berbasis token JWT (JSON Web Token). Untuk keamanan, digunakan werkzeug.security yang menyediakan fungsi hashing password seperti generate_password_hash dan check_password_hash untuk menyimpan password dengan aman. Penanganan waktu menggunakan datetime dan timedelta dari Pythonyang membantu mengelola tanggal, waktu, dan durasi. werkzeug.utils digunakan untuk keamanan file upload melalui fungsi secure_filename. Sistem database dihandle oleh SQLAlchemy yang memungkinkan query database dengan ORM, termasuk operator logika seperti or_.

2. Konfigurasi Database

```python
import os

class Config:
    # Database configuration
    SQLALCHEMY_DATABASE_URI = os.getenv(
        "DATABASE_URI",
        "mysql+mysqlconnector://root:@localhost:3306/eye_stunt"
    )
    SQLALCHEMY_TRACK_MODIFICATIONS = False

    # JWT configuration
    JWT_SECRET_KEY = os.getenv('JWT_SECRET_KEY', 'your-secret-key-here')  # Change this in production!
    JWT_ACCESS_TOKEN_EXPIRES = False  # Token will not expire

    SECRET_KEY = os.getenv('SECRET_KEY', 'your-secret-key-here')  # Change this in production!
```

2

Bagian ini merupakan konfigurasi database MySQL dan JSON Web Token untuk autentikasi. Konfigurasi database diatur melalui SQLALCHEMY_DATABASE_URI, yang menentukan koneksi ke database MySQL dengan nama eye_stunt. Jika tidak ada nilai yang diberikan melalui environment variable (DATABASE_URI), maka secara default aplikasi akan terhubung ke database dengan root tanpa password. Opsi SQLALCHEMY_TRACK_MODIFICATIONS dinonaktifkan (False) untuk menghindari overhead performa yang tidak diperlukan. Selanjutnya, konfigurasi JWT digunakan untuk mengamankan token autentikasi. JWT_SECRET_KEY adalah yang digunakan untuk menandatangani token JWT. Jika tidak disetel melalui environment variable, nilai default 'your-secret-key-here' akan digunakan. JWT_ACCESS_TOKEN_EXPIRES diatur ke False, yang berarti token tidak akan kadaluarsa. SECRET_KEY digunakan untuk melindungi session dan data sensitif.

3. Routes Registrasi

```python
@auth_bp.route('/register', methods=['POST'])
def register():
    data = request.get_json()

    # Validate required fields
    required_fields = ['name', 'email', 'password', 'birth_date']
    for field in required_fields:
        if field not in data:
            return jsonify({'error': f'Missing required field: {field}'}), 400

    # Check if user already exists
    if User.query.filter_by(email=data['email']).first():
        return jsonify({'error': 'Email already registered'}), 400

    try:
        # Create new user
        new_user = User(
            name=data['name'],
            email=data['email'],
            password=generate_password_hash(data['password']),
            birth_date=datetime.strptime(data['birth_date'], '%Y-%m-%d').date(),
            role=UserRole.USER
        )

        db.session.add(new_user)
        db.session.commit()

        # Create access token after successful registration
        access_token = create_access_token(identity=str(new_user.id))

        # Return complete user details and token in response
        response = {
            'message': 'User registered successfully',
            'access_token': access_token,
            'data': {
                'user': {
                    'id': new_user.id,
                    'name': new_user.name,
                    'email': new_user.email,
                    'role': new_user.role.value,
                    'is_verified': new_user.is_verified,
                    'birth_date': new_user.birth_date.strftime('%Y-%m-%d') if new_user.birth_date else None,
                    'created_at': new_user.created_at.strftime('%Y-%m-%d %H:%M:%S'),
                    'updated_at': new_user.updated_at.strftime('%Y-%m-%d %H:%M:%S') if new_user.updated_at else None
                }
            }
        }
```

Bagian diatas digunakan untuk mendaftarkan pengguna baru. Data yang dikirim oleh pengguna (seperti nama, email, password, dan tanggal lahir) diperiksa untuk memastikan semua field wajib terisi. Jika ada yang kurang, sistem akan mengembalikan pesan error dengan status kode 400 (Bad Request). Selanjutnya, sistem memeriksa apakah email yang didaftarkan sudah terdaftar sebelumnya di database. Jika email sudah ada, pengguna akan mendapatkan pesan error bahwa email sudah terdaftar. Jika email belum terdaftar, proses registrasi dilanjutkan dengan membuat akun baru. Password yang dimasukkan pengguna di-hash (dienkripsi) menggunakan generate_password_hash untuk keamanan sebelum disimpan ke database. Setelah data pengguna berhasil disimpan, sistem membuat access token menggunakan JWT (JSON Web Token) yang nantinya bisa digunakan untuk autentikasi. Token ini dikirim kembali ke pengguna bersama dengan detail profil seperti ID, nama, email, peran (role), status verifikasi, dan informasi lainnya dalam format JSON.

4. Routes Assesment

a. Routes Data Asessment

```python
# Route to view all questionnaires
@questionnaire_bp.route('/', methods=['GET'])
@jwt_required()
def view_questionnaires():
    try:
        # Fetch all questionnaires from the database
        questionnaires = Questionnaire.query.all()

        # Check if any questionnaires are available
        if not questionnaires:
            return jsonify({'message': 'No questionnaires found'}), 404

        # Return the questionnaires data
        return jsonify({
            'questionnaires': [q.to_dict() for q in questionnaires]
        }), 200
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

Bagian ini digunakan untuk melihat semua assesment yang tersimpan di database. Sistem akan mengambil assesment menggunakan Questionnaire.query.all(). Setiap assesment diubah ke format dictionary menggunakan metode to_dict(). Terdapat endpoint untuk memeriksa apakah pengguna yang akan login sudah pernah mengisi assesment atau belum. Sistem mengambil ID pengguna dari token JWT yang dikirim saat login. Kemudian, sistem mencari di database apakah ada assesment dengan ID pengguna tersebut. Jika ditemukan , sistem akan mengembalikan respons "User has already filled out the questionnaire" dengan status 200 (OK), menandakan bahwa pengguna sudah pernah mengisi assement.

b. Routes Mengisi Assesment

Pada bagian ini digunakan untuk membuat assesment baru bagi pengguna yang sedang login. Pertama, sistem mengambil ID pengguna dari token JWT dan memverifikasi apakah pengguna tersebut ada di database. Jika tidak ditemukan, sistem mengembalikan pesan error "User not found" dengan status 404. Sistem kemudian memvalidasi data yang dikirim, memastikan semua field wajib *(is_married, is_preparing_pregnancy, is_pregnant, dan app_purpose)* terisi. Jika ada yang kurang, sistem akan memberi tahu field mana yang belum diisi. Endpoint ini berguna untuk mengumpulkan informasi penting dari pengguna tentang status menikah, perencana kehamilan, remaja dewasa putri dan tujuan menggunakan aplikasi.

5. Routes Login

```python
@auth_bp.route('/login', methods=['POST'])
def login():
    data = request.get_json()

    # Validate required fields
    if not data.get('email') or not data.get('password'):
        response = {'error': 'Email and password are required'}
        print('Response:', response)  # Log to terminal
        return jsonify(response), 400

```

```
        # Verify user and password
        if not user or not check_password_hash(user.password, data['password']):
            response = {'error': 'Invalid email or password'}
            print('Response:', response)  # Log to terminal
            return jsonify(response), 401

        # Create access token
        access_token = create_access_token(identity=str(user.id))

        # Get user data including ID
        user_data = user.to_dict()

        response = {
            'message': 'Login successful',
            'access_token': access_token,
            'user': user_data
        }
        print('Response:', response)  # Log to terminal
        return jsonify(response), 200

    except Exception as e:
        response = {'error': str(e)}
        print('Response:', response)  # Log to terminal
        return jsonify(response), 500
```

Bagian ini digunakan untuk proses login pengguna. Pertama, sistem memeriksa apakah email dan password sudah diisi oleh pengguna. Jika ada yang kosong, sistem akan mengembalikan pesan error dengan status 400 (Bad Request). Sistem kemudian mencari pengguna berdasarkan email yang dimasukkan. Jika email tidak ditemukan atau password tidak cocok, sistem akan mengembalikan pesan "Invalid email or password" dengan status 401 (Unauthorized). Ini untuk mencegah akses dari orang yang tidak berhak. Jika email dan password benar, sistem akan membuat access token JWT yang berisi ID pengguna.

6. Routes Profile

a. Routes Update Profile

```
@auth_bp.route('/profile/update', methods=['PUT'])
@jwt_required()
def update_profile():
    """Update user profile information (name and email only)"""
    try:
        user_id = get_jwt_identity()
        user = User.query.get(user_id)

        if not user:
            return jsonify({'error': 'User not found'}), 404

        data = request.get_json()

        # Allowed fields for update
        allowed_fields = ['name', 'email']

        # Check if there are fields to update
        if not any(field in data for field in allowed_fields):
            return jsonify({'error': 'No valid fields to update'}), 400

        # Check if email is being updated and if it's already taken
        if 'email' in data and data['email'] != user.email:
            existing_user = User.query.filter_by(email=data['email']).first()
            if existing_user:
                return jsonify({'error': 'Email already taken'}), 400

        # Update name and email
        for field in allowed_fields:
            if field in data:
                setattr(user, field, data[field])

        db.session.commit()
```

Bagian ini digunakan untuk memperbarui profil pengguna khususnya untuk mengubah nama, email, dan tanggal lahir. Sistem akan memverifikasi identitas pengguna melalui token JWT yang dikirim. Proses update hanya memperbolehkan perubahan pada dua field tertentu, yaitu 'name', 'email' dan 'tanggal lahir'. Apabila tidak ada field valid yang ingin diupdate, sistem akan memberi respons error kepada pengguna. Untuk perubahan email, sistem melakukan pengecekan ketat. Jika email baru yang dimasukkan sudah terdaftar oleh pengguna lain, sistem akan menolak permintaan update dengan pesan "Email already taken". Setelah semua validasi berhasil, sistem akan memperbarui data dan menyimpannya ke database.

b. Routes Update Password

Bagian ini digunakan untuk untuk memperbarui password pengguna yang telah login. Route ini hanya dapat diakses oleh pengguna yang sudah terautentikasi melalui JWT, yang dibuktikan dengan adanya dekorator @jwt_required(). Ketika route ini dipanggil, sistem akan mengambil ID pengguna dari token JWT menggunakan get_jwt_identity(), lalu mencari data pengguna tersebut di database. Jika pengguna tidak ditemukan, server akan mengembalikan pesan error 404. Selanjutnya, server akan memproses data JSON yang dikirimkan oleh pengguna dan memeriksa apakah semua field yang dibutuhkan tersedia, yaitu current_password, new_password, dan confirm_password.

7. Routes Artikel Kesehatan

```python
@auth_bp.route('/artikel/<int:id>', methods=['GET'])
@jwt_required()
def get_artikel_detail(id):
    """Ambil detail artikel berdasarkan ID"""
    try:
        current_user_id = int(get_jwt_identity())  # kalau mau tracking user

        artikel = HealthArticle.query.get_or_404(id)
        return jsonify({
            'artikel': {
                'id': artikel.id,
                'judul': artikel.judul,
                'penulis': artikel.penulis,
                'isi': artikel.isi,
                'tanggal_publikasi': artikel.tanggal_publikasi.strftime('%Y-%m-%d'),
                'gambar': artikel.gambar
            }
        }), 200

    except Exception as e:
        logging.error(f"Error fetching article ID {id}: {str(e)}")
        logging.debug("Traceback:\n" + traceback.format_exc())
        return jsonify({'message': f'Error fetching article: {str(e)}'}), 500
```

Bagian ini digunakan untuk mengambil detail artikel kesehatan berdasarkan ID-nya dan hanya bisa diakses oleh pengguna yang telah login, karena dilindungi oleh @jwt_required(). Ketika diakses, sistem akan mengambil ID pengguna dari token JWT . Artikel akan dicari di database menggunakan HealthArticle.query.get_or_404(id), yang otomatis mengembalikan error 404 jika artikel tidak ditemukan. Jika berhasil, data artikel seperti judul, penulis, isi, tanggal publikasi, dan gambar akan dikirimkan dalam format JSON. Jika terjadi kesalahan, sistem akan mencatat log error dan mengembalikan pesan error 500.

8. Routes Chat

a. Routes Melihat Pesan

```python
@chat_bp.route('/chat/<int:consultation_id>/messages', methods=['GET'])
@jwt_required()
def get_chat_messages(consultation_id):
    try:
        user_id = get_jwt_identity()

        # Check if consultation exists
        consultation = Consultation.query.get(consultation_id)
        if not consultation:
            return jsonify({'error': 'Consultation not found'}), 404

        # Check if consultation is in a valid state for chat
        if consultation.status not in [ConsultationStatus.CONFIRMED, ConsultationStatus.COMPLETED]:
            return jsonify({'error': 'Chat unavailable for this consultation status'}), 400

        # Mark messages as read for this user
        messages_read = ChatMessage.mark_as_read(consultation_id, user_id)

        # Get all messages
        messages = ChatMessage.get_chat_history(consultation_id)

        # Convert messages to dict with sender details
        message_list = []
        for msg in messages:
            message_dict = {
                'id': msg.id,
                'consultation_id': msg.consultation_id,
                'sender_id': msg.sender_id,
                'sender': {
                    'id': msg.sender.id,
                    'name': msg.sender.name,
                    'role': msg.sender.role.value,
                    'email': msg.sender.email
                },
                'content': msg.content,
                'created_at': msg.created_at.isoformat(),
                'status': msg.status.value
            }
            message_list.append(message_dict)

        return jsonify({
            'success': True,
            'messages': message_list,
            'messages_count': len(messages),
            'messages_read': messages_read
        }), 200

    except Exception as e:
        print(f"Error getting chat messages: {str(e)}")
        return jsonify({
            'success': False,
            'error': str(e)
        }), 500
```

Bagian ini digunakan untuk mengambil semua pesan chat dalam suatu sesi konsultasi berdasarkan consultation_id. Sistem mengambil ID pengguna dari token JWT, lalu memeriksa apakah konsultasi dengan ID yang diberikan benar-benar ada. Jika tidak ditemukan, akan dikembalikan respons error 404. Selanjutnya, sistem memvalidasi apakah status konsultasi tersebut berada dalam kondisi yang mengizinkan penggunaan fitur chat, yaitu hanya jika statusnya CONFIRMED atau COMPLETED. Jika status tidak sesuai, maka respons 400 akan dikirimkan.

b. Routes Membuat Pesan

```python
@chat_bp.route('/chat/<int:consultation_id>/messages', methods=['POST'])
@jwt_required()
def send_message(consultation_id):
    try:
        user_id = get_jwt_identity()
        data = request.get_json()

        # Check if required fields are present
        if 'content' not in data or not data['content'].strip():
            return jsonify({'error': 'Message content is required'}), 400

        # Check if consultation exists
        consultation = Consultation.query.get(consultation_id)
        if not consultation:
            return jsonify({'error': 'Consultation not found'}), 404

        # Check if consultation is in a valid state for chat
        if consultation.status not in [ConsultationStatus.CONFIRMED, ConsultationStatus.COMPLETED]:
            return jsonify({'error': 'Chat unavailable for this consultation status'}), 400

        # Create new message
        new_message = ChatMessage(
            consultation_id=consultation_id,
            sender_id=user_id,
            content=data['content'],
            status=MessageStatus.SENT
        )

        db.session.add(new_message)
        db.session.commit()

        return jsonify({
            'success': True,
            'message': 'Message sent successfully',
            'chat_message': new_message.to_dict()
        }), 201

    except Exception as e:
        db.session.rollback()
        return jsonify({
            'success': False,
            'error': str(e)
        }), 500
```

Bagian ini digunakan untuk mengirim pesan chat dalam sesi konsultasi tertentu. Hanya pengguna yang telah login yang dapat mengaksesnya. Saat endpoint ini dipanggil, sistem akan mengambil ID pengguna dari token JWT dan memeriksa apakah data pesan (content) tersedia dan tidak kosong. Selanjutnya, sistem akan memastikan bahwa konsultasi dengan ID tersebut ada dan berada dalam status yang memperbolehkan chat (CONFIRMED atau COMPLETED). Jika semua validasi lolos, pesan baru akan dibuat, disimpan ke database, dan dikembalikan ke pengguna dalam format JSON sebagai konfirmasi. Jika terjadi kesalahan, sistem akan membatalkan transaksi dan mengembalikan pesan error.

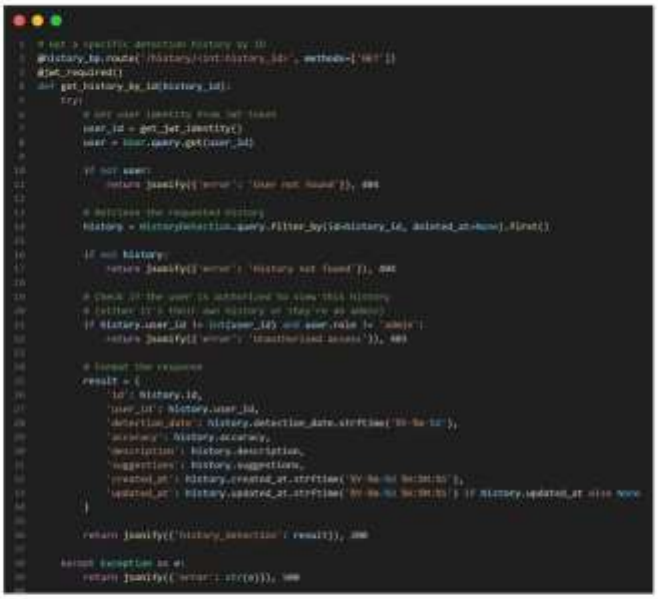
9. Routes History

a. Routes History Upload

```python
@history_bp.route('/history/upload', methods=['POST'])
@jwt_required()
def upload_detection():
    user_id = get_jwt_identity()
    file = request.files.get('image')

    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        os.makedirs(UPLOAD_FOLDER, exist_ok=True)
        filepath = os.path.join(UPLOAD_FOLDER, filename)
        file.save(filepath)

        # Get the current time (local server time)
        now = datetime.now()

        detection = HistoryDetection(
            user_id=user_id,
            detection_date=now.date(),
            created_at=now,
            updated_at=now,
            accuracy=float(request.form['accuracy']),
            description=request.form['description'],
            suggestions=request.form['suggestions'],
            image_filename=filename
        )
        db.session.add(detection)
        db.session.commit()

        return jsonify({'message': 'Deteksi berhasil disimpan'}), 201

    return jsonify({'error': 'Gambar tidak valid'}), 400
```

Bagian ini digunakan untuk mengunggah hasil deteksi gambar. Ketika pengguna mengirimkan gambar, sistem akan memeriksa apakah file gambar valid, lalu menyimpannya ke folder khusus di server. Setelah itu, data deteksi seperti tingkat akurasi, deskripsi, saran, dan nama file gambar akan disimpan ke database bersama dengan informasi waktu saat deteksi dilakukan. Jika berhasil, sistem akan mengirimkan respons bahwa deteksi telah tersimpan. Jika file gambar tidak valid atau tidak ada, maka akan dikembalikan pesan error.

b. Routes Detail History



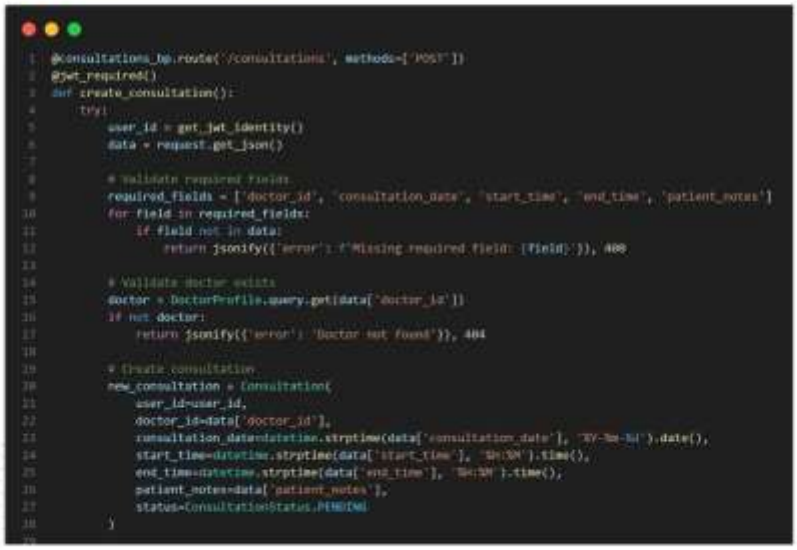
Bagian ini digunakan untuk mengambil detail riwayat deteksi tertentu berdasarkan ID-nya. Setelah mengambil identitas pengguna dari token JWT, sistem memverifikasi apakah pengguna tersebut ada. Selanjutnya, sistem mencari data riwayat deteksi yang sesuai dengan ID dan belum dihapus. Jika ditemukan, sistem akan memeriksa apakah pengguna tersebut adalah pemilik riwayat tersebut. Jika diizinkan, data riwayat deteksi seperti tanggal, akurasi, deskripsi, dan saran akan dikembalikan dalam format JSON. Jika tidak ditemukan atau akses tidak sah, sistem akan mengembalikan pesan error yang sesuai.

c. Routes History List

Bagian ini digunakan oleh pengguna yang sudah login untuk melihat seluruh riwayat deteksi milik user. Setelah sistem memverifikasi identitas pengguna dari token JWT, data pengguna akan diambil dari database. Jika pengguna ditemukan, sistem juga akan mengambil data dari assesment terbaru untuk mengetahui apakah pengguna sedang hamil atau dalam persiapan kehamilan. Kemudian, sistem mengambil semua riwayat deteksi milik pengguna yang belum dihapus dan mengurutkannya berdasarkan tanggal terbaru. Setiap riwayat akan diformat dengan informasi seperti tanggal deteksi, akurasi, deskripsi, saran, dan URL gambar hasil deteksi. Hasil akhirnya akan dikembalikan dalam format JSON bersama dengan nama dan ID pengguna.

10. Routes Konsultasi

a. Routes Pemesanan Konsultasi



```
        db.session.add(new_consultation)
        db.session.commit()

        # Convert to dictionary for JSON response
        consultation_dict = new_consultation.to_dict()

        # Include the consultation_id in the response using BOTH formats for maximum compatibility
        return jsonify({
            'success': True,
            'message': 'Consultation created successfully',
            'consultation': consultation_dict,
            'consultation_id': new_consultation.id,
            'consultationId': new_consultation.id  # Add this field for compatibility with Flutter code
        }), 201
    except Exception as e:
        db.session.rollback()
        print(f"Error creating consultation: {str(e)}")
        return jsonify({
            'success': False,
            'error': str(e)
        }), 500
```

Bagian ini digunakan untuk membuat konsultasi baru antara pasien dan dokter. Setelah itu, data konsultasi seperti doctor_id, consultation_date, start_time, end_time diambil dari request body dan divalidasi satu per satu untuk memastikan tidak ada yang kosong. Kemudian, sistem memeriksa apakah dokter dengan ID yang diberikan benar-benar ada di database. Jika semua validasi berhasil, maka sistem akan membuat objek konsultasi baru dengan status awal PENDING, lalu menyimpannya ke database. Route ini memastikan proses pembuatan konsultasi berjalan aman dan terstruktur.

b. Routes Konsultasi Terbayar

```python
@consultations_bp.route('/consultations/paid', methods=['GET'])
@jwt_required()
def get_paid_consultations():
    try:
        user_id = get_jwt_identity()
        user = User.query.get(user_id)

        # Build the base query joining consultations with their payments and doctor info
        query = (
            db.session.query(Consultation, DoctorProfile, User, Payment)
            .join(Payment, Payment.consultation_id == Consultation.id)
            .join(DoctorProfile, DoctorProfile.id == Consultation.doctor_id)
            .join(User, User.id == DoctorProfile.user_id)
            .filter(Payment.status == PaymentStatus.PAID)
        )

        # Filter by user role
        if user.is_doctor():
            # Get doctor's profile
            doctor_profile = DoctorProfile.query.filter_by(user_id=user_id).first()
            if not doctor_profile:
                return jsonify({'error': 'Doctor profile not found'}), 404
            query = query.filter(Consultation.doctor_id == doctor_profile.id)
        else:
            # Regular user - only show their own consultations
            query = query.filter(Consultation.user_id == user_id)

        # Optional date range filtering
        start_date = request.args.get('start_date')
```

```
        if start_date:
            start_date = datetime.strptime(start_date, '%Y-%m-%d').date()
            query = query.filter(Consultation.consultation_date >= start_date)

        if end_date:
            end_date = datetime.strptime(end_date, '%Y-%m-%d').date()
            query = query.filter(Consultation.consultation_date <= end_date)

        # Execute query and get results
        results = query.order_by(Consultation.consultation_date.desc()).all()

        # Format the response with complete information
        consultations_list = []
        for consultation, doctor_profile, doctor_user, payment in results:
            # Get the patient user information
            patient_user = User.query.get(consultation.user_id)

            # Create a comprehensive dictionary with all data
            consultation_data = {
                # Consultation details
                **consultation.to_dict(),

                # Doctor information
                'doctor': {
                    **doctor_profile.to_dict(),
                    'name': doctor_user.name,
                    'email': doctor_user.email
                },

                # Patient information
                'patient': patient_user.to_dict() if patient_user else None,

                # Payment information
                'payment': payment.to_dict()
            }

            consultations_list.append(consultation_data)

        return jsonify({
            'success': True,
            'paid_consultations': consultations_list,
            'count': len(consultations_list)
        }), 200
    except Exception as e:
        print(f"Error getting paid consultations: {str(e)}")
        return jsonify({
            'success': False,
            'error': str(e)
        }), 500
```

Bagian ini digunakan untuk mengambil daftar konsultasi yang sudah dibayar oleh pengguna yang sedang login. Saat route ini diakses, sistem akan mengecek apakah pengguna adalah dokter atau pasien. Jika pengguna adalah dokter, maka sistem akan menampilkan semua konsultasi berstatus "PAID" yang berkaitan dengan dokter. Jika pengguna adalah pasien, maka hanya konsultasi mereka sendiri yang ditampilkan. Data yang diambil tidak hanya mencakup detail konsultasi, tetapi juga informasi dokter, pasien, dan pembayaran. Selain itu, route ini juga mendukung filter berdasarkan rentang tanggal melalui parameter start_date dan end_date di URL.

15

11. Routes Payment

16

Bagian ini digunakan untuk membuat pembayaran konsultasi oleh pengguna yang sudah login. Ketika pengguna mengirim permintaan POST dengan data seperti consultation_id, payment_method, dan payment_provider, sistem akan memverifikasi apakah konsultasi tersebut ada, dimiliki oleh pengguna, dan belum dibayar. Jika valid, sistem mengecek apakah slot waktu konsultasi masih tersedia dan belum diambil orang lain. Selanjutnya, sistem membuat entri pembayaran di database, menghitung total biaya berdasarkan tarif dokter, dan mengatur batas waktu pembayaran selama 24 jam. Sistem lalu menyiapkan data transaksi dan mengirimkannya ke Midtrans untuk memproses pembayaran. Jika berhasil, sistem menyimpan transaction_id dari Midtrans dan memberikan URL pembayaran (jika tersedia) kembali ke pengguna dalam respons JSON. Jika terjadi kesalahan, transaksi dibatalkan dan pesan error dikembalikan.

12. Routes Deteksi

Bagian ini digunakan untuk mendeteksi anemia dari gambar mata (khususnya konjungtiva) yang diunggah oleh pengguna. Server memeriksa apakah file gambar dikirim dan apakah formatnya valid (JPG, PNG, atau HEIC). Setelah itu, gambar disimpan lalu diperiksa apakah area mata terlihat jelas dan terbuka. Jika valid, gambar akan dipotong secara otomatis ke bagian konjungtiva dan diproses oleh model deteksi anemia. Hasil prediksi dikategorikan "Normal" atau "Suspek Anemia" dan diberikan saran kesehatan. Informasi hasil deteksi kemudian disimpan ke database sebagai riwayat pengguna, dan gambar yang sudah dipotong disediakan dalam bentuk URL untuk ditampilkan.

13. Routes Chatbot

```python
from flask import Blueprint, request, jsonify
from app.controllers.chatbot import get_response,load_chatbot_resources,predict_class
from flask import Blueprint, jsonify, request

chatbot_bp = Blueprint('chatbot', __name__)
@chatbot_bp.route('/chat', methods=['POST'])
def chat():
    print("\n==================================")
    print("💬 New Chatbot Request")
    print("==================================")
    try:
        # Get message from request
        data = request.get_json()
        if not data or 'message' not in data:
            return jsonify({'error': 'No message provided'}), 400
        message = data['message']
        print(f"Received message: {message}")
        # Load resources
        model, words, classes, intents = load_chatbot_resources()
        if None in (model, words, classes, intents):
            return jsonify({'error': 'Failed to load chatbot resources'}), 500
        # Get prediction
        ints = predict_class(message, model, words, classes)
        # Get response
        response = get_response(intents, ints)
        result = {
            'response': response,
            'intent': ints[0]['intent'] if ints else 'unknown',
            'confidence': float(ints[0]['probability']) if ints else 0.0
        }

        print("\n=== Response ===")
        print(f"Intent: {result['intent']}")
        print(f"Confidence: {result['confidence']:.2f}")
        print(f"Response: {result['response']}")
        print("==================================\n")

        return jsonify(result), 200

    except Exception as e:
        error_msg = {'error': str(e)}
        print("\n❌ Error Processing Request:")
        print(f"Error type: {type(e).__name__}")
        print(f"Error message: {str(e)}")
        print("==================================\n")
        return jsonify(error_msg), 500

```

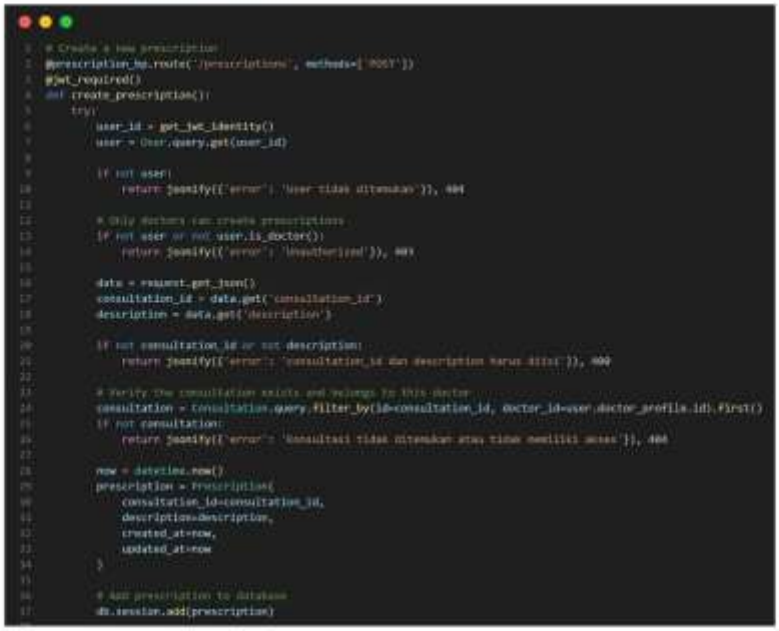
Bagian ini digunakan untuk menangani permintaan chat dari pengguna ke chatbot. Saat pengguna mengirim pesan melalui metode POST, sistem akan memeriksa apakah pesan tersedia. Jika tidak, akan dikembalikan pesan error. Jika pesan valid, server memuat sumber daya chatbot seperti model, daftar kata, kelas, dan intents. Setelah itu, pesan akan diproses untuk memprediksi maksud (intent) pengguna menggunakan model yang telah dilatih. Berdasarkan hasil prediksi tersebut, chatbot akan menghasilkan respons yang sesuai. Hasil berupa respons chatbot, intent yang terdeteksi, dan tingkat kepercayaan (confidence) akan dikembalikan dalam format JSON.

14. Routes Jadwal Konsultasi

```python
@doctor_schedule_bp.route('/doctors/schedules', methods=['POST'])
@jwt_required()
def create_doctor_schedule():
    try:
        user_id = get_jwt_identity()
        user = User.query.get(user_id)

        if not user or not user.is_doctor():
            return jsonify({'error': 'Unauthorized'}), 403

        data = request.get_json()

        doctor_profile = DoctorProfile.query.filter_by(user_id=user_id).first()
        if not doctor_profile:
            return jsonify({'error': 'Doctor profile not found'}), 404

        required_fields = ['day_of_week', 'is_available']
        for field in required_fields:
            if field not in data:
                return jsonify({'error': f'Missing required field: {field}'}), 400

        try:
            day_of_week = DayOfWeek[data['day_of_week'].upper()]
        except KeyError:
            valid_days = [e.name for e in DayOfWeek]
            return jsonify({
                'error': f'Invalid day_of_week. Valid values: {", ".join(valid_days)}'
            }), 400

        existing_schedule = DoctorSchedule.query.filter_by(
            doctor_id=doctor_profile.id,
            day_of_week=day_of_week
        ).first()

        if existing_schedule:
            return jsonify({'error': 'Schedule for this day already exists'}), 400

        schedule = DoctorSchedule(
            doctor_id=doctor_profile.id,
            day_of_week=day_of_week,
            is_available=data.get('is_available', True)
        )

        db.session.add(schedule)
        db.session.commit()

        return jsonify({
            'success': True,
            'message': 'Schedule created successfully',
            'schedule': schedule.to_dict()
        }), 201

    except Exception as e:
        db.session.rollback()
        logging.error(f"Error creating doctor schedule: {str(e)}")
        return jsonify({
            'success': False,
            'error': str(e)
        }), 500
```

Bagian ini digunakan untuk membuat jadwal praktik dokter. Hanya pengguna yang sudah login dan memiliki peran sebagai dokter yang dapat mengakses. Sistem akan mengambil ID pengguna dari token JWT, lalu memastikan bahwa pengguna adalah dokter dan valid. Setelah itu, sistem memeriksa apakah data input berisi informasi hari praktik (day_of_week) dan status ketersediaan (is_available). Jika data valid, sistem akan mengecek apakah jadwal pada hari tersebut sudah ada. Jika belum, sistem akan menyimpan jadwal baru ke database. Jika berhasil, akan dikembalikan respons JSON berisi data jadwal yang baru dibuat. Namun jika terjadi kesalahan, sistem akan membatalkan transaksi database dan mengembalikan pesan error.

15. Routes Advice Dokter



```python
# Create a new prescription
@prescription_bp.route('/prescriptions', methods=['POST'])
@jwt_required()
def create_prescription():
    try:
        user_id = get_jwt_identity()
        user = User.query.get(user_id)

        if not user:
            return jsonify({'error': 'User tidak ditemukan'}), 404

        # Only doctors can create prescriptions
        if not user or not user.is_doctor():
            return jsonify({'error': 'Unauthorized'}), 403

        data = request.get_json()
        consultation_id = data.get('consultation_id')
        description = data.get('description')

        if not consultation_id or not description:
            return jsonify({'error': 'consultation_id dan description harus diisi'}), 400

        # Verify the consultation exists and belongs to this doctor
        consultation = Consultation.query.filter_by(id=consultation_id, doctor_id=user.doctor_profile.id).first()
        if not consultation:
            return jsonify({'error': 'Konsultasi tidak ditemukan atau tidak memiliki akses'}), 404

        now = datetime.now()
        prescription = Prescription(
            consultation_id=consultation_id,
            description=description,
            created_at=now,
            updated_at=now
        )

        # Add prescription to database
        db.session.add(prescription)
```

Bagian ini digunakan oleh dokter untuk membuat advice berdasarkan konsultasi pasien. Sistem memverifikasi identitas pengguna dan memastikan bahwa pengguna tersebut adalah seorang dokter. Kemudian, data consultation_id dan description dari body request diperiksa—keduanya wajib diisi. Setelah itu, sistem memastikan bahwa konsultasi yang dimaksud memang milik dokter tersebut. Jika semuanya valid, sistem akan membuat entri resep baru, menyimpan ke database, dan secara otomatis mengubah status konsultasi menjadi "selesai". Respon yang dikembalikan berisi informasi resep dan status konsultasi, dan jika terjadi kesalahan, proses akan dibatalkan dan error akan ditampilkan.

Lampiran 5. Sertifikat HKI

REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka pelindungan 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	:	EC002025076723, 26 Juni 2025
Pencipta		
Nama	:	**Lulu Nadhiatun Aniss, Dwi Intan Af'idah, S.T., M.Kom. dkk**
Alamat	:	Jalan Kapten Samadikun, RT.01/ RW.01, No.6, Desa Sumurpanggang, Margadana, Kota Tegal, Jawa Tengah, 52141
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**
Alamat	:	Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**Aplikasi Deteksi Anemia Melalui Konjugtiva Mata Untuk Perempuan Usia Produktif Sebagai Pencegahan Stunting**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	24 Juni 2025, di Kota Tegal
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	:	000916984

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
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko,SH.,MH.
NIP. 196912261994031001

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

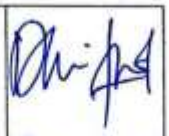
Lampiran 6. Lembar Bimbingan

D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Lulu Nadhiatun Anisa
NIM : 21090066
No.Ponsel : 0895384257068
Judul TA : Aplikasi Deteksi Dini Anemia Melalui Konjugtiva Mata Untuk Perempuan Usia Produktif Sebagai Upaya Pencegahan Stunting
Dosen Pembimbing I : Dwi Intan Af'idah, S.T., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	21/3 - 2025	Konsep	Perbaikan judul dan perbaikan model	
2.	17/4 - 2025	Aplikasi	Implementasi model dan fokus penyelesaian fitur utama	
3.	9/5 - 2025	Aplikasi	Selesaikan fitur - fitur yang dapat digunakan pengguna	
4.	23/5 - 2025	Aplikasi	Lanjutkan fitur untuk role dokter -Pertimbangkan solusi untuk mata yang tidak terlihat konjungtiva	
5	2/6 - 2025	Aplikasi	-Guideline untuk mengatur konjungtiva harus lebih jelas -lanjutkan fitur role dokter	

6.	13/6 - 2025	Aplikasi	-aplikasi sudah cukup -persiapkan dokumen pengajuan hak cipta	
7.	19/6 - 2025	Laporan	- Tabel untuk sample dataset chatbot	
8.	25/6 - 2025	laporan	-Bagian awal laporan dirapikan - saran diperbaiki	
9.	26/6 - 2025	laporan	ACC	

Tegal, 26 Juni 2025
Dosen Pembimbing I

Dwi Intan Af'idah, S.T., M.Kom.
NIPY. 11.020.470

SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : **Lulu Nadhiatun Anisa**
Nim : **21090066**
No. Ponsel : **0895384257068**
Judul : **Aplikasi Deteksi Anemia Melalui Konjugtiva Mata Untuk Perempuan Usia Produktif Sebagai Upaya Pencegahan Stunting**
Dosen Pembimbing II : **Hepatika Zidny Ilmadina, S.Pd., M.Kom.**

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	21/03/2025	Aplikasi	- Model deteksi anemia -Judul Laporan	
2.	21/04/2025	Aplikasi	Penambahan fitur resep obat	
3.	19/05/2025	Aplikasi	Penyempurnaan aplikasi.	
4.	2/06/2025	Aplikasi	Penyempurnaan aplikasi.	
5.	17/06/2025	HKI	Manual Book dan Dokumen Teknis	
6.	20/06/2025	Laporan Bab 1	-	
7.	15/06/2025	Laporan Bab 2	Pengujian Aplikasi	
8.	26/06/2025	Laporan Bab 3	Acc Laporan	

Tegal, 26 Juni 2025
Dosen Pembimbing II,

Hepatika Zidny Ilmadina, S.Pd., M.Kom.
NIPY. 09.015.225