

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Dosen Pembimbing 1

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Wafah Lailatul Istianah
NIM : 22090005
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 Muda Tk.I - III/b

Pada hari ini selase tanggal 12 Mei 2026 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat adanya peningkatan progres dari proyek skripsi yang dilaporkan setiap satu pekan. Jika Pihak Pertama tidak memenuhi syarat tersebut, maka Pihak Kedua berhak memutuskan kesepakatan ini. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 12 Mei 2026

Pihak Pertama



Wafah Lailatul Istianah

Pihak Kedua



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

Mengetahui
Ketua Program Studi Sarjana Terapan
Teknik Informatika


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

Lampiran 2. Surat Kesepakatan Bimbingan Dosen Pembimbing

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Wafah Lailatul Istianah
NIM : 22090005
Program Studi : Sarjana terapan Teknik Informatika

Pihak Kedua

Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
Status : Dosen
NUPTK : 2154773674130253
Jabatan Fungsional : Dosen Tetap
Pangkat/Golongan : -

Pada hari ini Rabu tanggal 13 Mei 2026 telah dibuat kesepakatan antara **Pihak Pertama (Mahasiswa)** dan **Pihak Kedua (Dosen Pembimbing)** bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir bagi Pihak Pertama.

Sebagai bagian dari proses penyelesaian Tugas Akhir, Pihak **Pertama wajib melakukan bimbingan secara rutin sebanyak 1 (satu) kali dalam setiap minggu** kepada Pihak Kedua. Bimbingan tersebut **tidak dapat dirapel atau digabungkan** pada minggu berikutnya apabila pada minggu berjalan tidak dilaksanakan.

Setiap kegiatan bimbingan wajib disertai dengan **presentasi perkembangan Tugas Akhir** yang telah dipersiapkan oleh Pihak Pertama sesuai dengan ketentuan yang telah ditetapkan. Selain itu, selama proses penyusunan Tugas Akhir, **Pihak Pertama wajib melaksanakan bimbingan dengan jumlah minimal 10 (sepuluh) kali pertemuan** yang tercatat sebagai bukti proses pembimbingan. Adapun waktu dan tempat pelaksanaan bimbingan disepakati bersama oleh kedua belah pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran dan tanggung jawab oleh kedua belah pihak guna mendukung kelancaran proses penyelesaian Tugas Akhir.

Tegal, 13 Mei 2026

Pihak Pertama


Wafah Lailatul Istianah

Pihak Kedua


Arif Hidayah, S.Tr.Kom., M.Tr.Kom.

Mengetahui
Ketua Program Studi Teknik Informatika


Dyah Apriliani, S.Tr.Kom., M.Kom.
NIPY 09.015/225

Lampiran 3. Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta.

1. N a m a : **Wafah Lailatul Istianah**
Kewarganegaraan : **Indonesia**
Alamat : **Desa Pulosari Rt.01/Rw.05 Kecamatan Brebes , Kabupaten Brebes, Jawa tengah**
2. N a m a : **Dwi Intan Af'idah, S.T., M.Kom.**
Kewarganegaraan : **Indonesia**
Alamat : **Desa Grinting RT/RW 03/01, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah 52253.**
3. N a m a : **Arif Hidayah, S.Tr.Kom., M.Tr.Kom**
Kewarganegaraan : **Indonesia**
Alamat : **JL. BULU EMAS No.8, Desa Lawatan, Kecamatan Dukuhturi, Kabupaten Tegal, Jawa Tengah**

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : **Program Komputer**
Berjudul : **Ranjang Bangun Aplikasi Analisis Resiko Dan Pencegahan Penyakit Jantung Sesuai Pola Gaya Hidup Menggunakan Rule Based**
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam 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 14 Juni 2026



Wafan Lailatul Istianan
Pemegang Hak Cipta *

Dwi Intan Afidah, S.T., M.Kom
Pemegang Hak Cipta *

Arif Hidayat, S.Tr.Kom., MTr.Kom
Pemegang Hak Cipta *

* Semua pemegang hak cipta agar menandatangani di atas materai (untuk penulis pertama)

Lampiran 4. Surat Permohonan Izin Observasi

 **Universitas
Harkat Negeri**

Fakultas Sekolah Vokasi
Program Studi D-4 Teknik Informatika

Nomor : 95.03/TI.HN/VII/2026
Lampiran : -
Hal : Permohonan Ijin Observasi
Kepada :
Yth. : **Kepala RSUD Kardinah**
Di Tempat

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Wafah Lailatul Istianah
NIM : 22090005
no hp : 085748592862
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul
"Rancang Bangun aplikasi analisis resiko dan pencegahan penyakit jantung sesuai pola gaya hidup menggunakan rule based". Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar dapat memperoleh data, keterangan, dan bahan yang diperlukan.

Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.

Tegal, 22 Juli 2026
Ka Prodi S.Tr. Teknik Informatika,


Dyan Apriliani, S.T., M.Kom
NIPY : 091015.225

Program Studi D-4 Teknik Informatika, Fakultas Sekolah Vokasi
Gedung D Ruang D2.15 Kampus Mataram – Jalan Mataram No.9 Kota Tegal
WhatsApp: +62-896-2520-0020 | Surel: userprodi@harkatnegeri.ac.id

Lampiran 5. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. N a m a : Wafah Lailatul Istianah
Kewarganegaraan : Indonesia
Alamat : Desa Pulosari Rt.01/Rw.05 Kecamatan Brebes, Kabupaten Brebes, Jawa tengah
2. N a m a : Dwi Intan Afidah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Grinting RT/RW 03/01, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah 52253.
3. N a m a : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : JL. BULU EMAS No.8, Desa Lawatan, Kecamatan Dukuhhuri, Kabupaten Tegal, Jawa Tengah

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

N a m a : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Universitas Harkat Negeri
Alamat : Jl. Mataram No.9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Rancang Bangun Aplikasi Analisis Resiko Dan Pencegahan Penyakit Jantung Sesuai Pola Gaya Hidup Menggunakan Rule Based ". 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, 14 Juni 2026

Pemegang Hak Cipta
 Universitas P3M
Harkat Negeri
(Sharfina Febbi Handayani, M.Kom.)


(Wafah Lailatul Istianah)


(Dwi Intan Afidah, S.T., M.Kom.)


(Arif Hidayah, S.Tr.Kom., M.Tr.Kom)

Lampiran 6. Dokumentasi Observasi



Lampiran 7. Dokumentasi Validasi Aplikasi



Lampiran 8. Surat Izin Validasi Dokter RS Kardinah Tegal

INSTALASI DIKLAT

Hal : Ijin Studi Penelitian

Tegal, 11 Agustus 2026

Kepada Yth. :

Koordinator PDE

Dengan hormat,

Menindak lanjuti disposisi Direktur RSUD Kardinah pada surat dari

Nama Institusi : Universitas Harkat Negeri

Nomor Surat : 60.03/TI.HN/III/2026

Perihal : **Permohonan Ijin Observasi**

bersama ini kami mohon dapat diperkenankan kepada :

NAMA : WAFAH LAILATUL ISTIANAH

Program Studi : D-IV Teknik Informatika

untuk mendapatkan data sesuai dengan keperluan dalam rangka penyusunan
KTI/Skripsi/tesis dengan Judul :

*"RANCANG BANGUN APLIKASI ANALISIS RESIKO DAN PENCEGAHAN PENYAKIT
JANTUNG SESUAI POLA GAYA HIDUP"*

Adapun persyaratan kelengkapan administrasi di RSUD Kardinah sudah dipenuhi.

Demikian untuk menjadikan maklum dan atas kerjasamanya terima kasih.

Ka. Sub. Bagian Kepegawaian & Pengembangan SDM
RSUD Kardinah Kota Tegal



Lampiran 9. Surat Lembar Bimbingan 1



**D IV TEKNIK INFORMATIKA
UNIVERSITAS HARKAT NEGERI**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Wafah Lailatul Istianah
 NIM : 22090005
 No. : 085748592862
 Ponsel
 Judul TA : Sistem Analisis Risiko Kesehatan Jantung Berdasarkan Pola Gaya Hidup Anak Muda Menggunakan Smartband dan Machine Learning

Dosen Pembimbing I : Dwi Intan Afidah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	26/05-26	Aplikasi	- Integrasi dg smartband	
2.	8/6-26	Aplikasi	Perbanyak fitur yang dapat membantu pengguna alarm	
3.	22/6-26	Aplikasi	- pemantauan olahraga - manajemen stress - Analisis risiko penyakit jantung	



Dipindai dengan CamScanner

4.	26/6-20	Aplikasi	Perum verifikasi ahli	
5.	2/7-26	Aplikasi	- Tunjukkan tabel y/ skor dari sistem pakar - Suggestion dari Analisis Risiko - Suggestion akan jadi baseline /target pada aktivitas fisik - Report aktivitas fisik	
6.	13/7-26	Aplikasi	lanjutkan hak cipta	
7.	20/7-26	Laporan	Perbaiki sesuai catatan	
8.	28/7-26	Laporan	ACC	

Tegal, ~~2020~~ 2026
Dosen Pembimbing I


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

Lampiran 10. Surat Lembar Bimbingan 2



D IV TEKNIK INFORMATIKA
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Wafah Lailatul Istianah
NIM : 22090005
No. Ponsel : 085748592862
Judul TA : Sistem Analisis dan Pencegahan Risiko Kesehatan Jantung Berdasarkan Pola Gaya Hidup Anak Muda

Dosen Pembimbing II : Arif Hidayah, S.Tr.Kom., M.Tr.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	6/6/26	PPP	Perbaiki	
2	3/7/26	PPP	Perbaiki Perbaikan bag	
3	6/2/26	PPP	- Review pd analisis risiko - ~~hasil~~ output diinput ke komputer	
4	8/7/26	PPP	- Online ke server - the form gplay - perbaiki bag	

CS Dipindai dengan CamScanner

5	13/7/26	PRR	- OK - Lantai Hekel. - Lantai Lupa	
6	14/7/26	Lupa	Lantai Kaki II	
7	12/7/26	Lupa	- Perbaikan Lantai - Sumbat - Atura Induk	
8	27/7/26	Lupa	- Lantai - Sisi	
		PRR	- Onboarding	
9	28/7/26	APP	- Perbaikan display	

Tegal, Mei 2026
Dosen Pembimbing II


Arif Hidayah, S.Tr.Kom., M.Tr.Kom.
NIPY. 10.024.548

HEARTIFY

Your Heart, Your Healthy
Future



MANUAL BOOK

WAFAH LAILATUL ISTIANAH

DAFTAR ISI

PENDAHULUAN	H-3
PENGUNAAN APLIKASI	H-4
1. REGISTER	H-4
2. LOGIN	H-5
3. HOME	H-5
4. PROFILE	H-6
5. ANALISIS RESIKO JANTUNG	H-8
6. HASIL ANALISIS	H-9
7. RIWAYAT ANALISIS	H-10
8. AKTIVITAS OLAHRAGA	H-10
9. RIWAYAT OLAHRAGA	H-11
10. MANEJEMEN STRESS	H-12
11. FITUR PENGINGAT	H-15

PENDAHULUAN

Selamat datang di ***Heartify*** , sebuah aplikasi analisis risiko penyakit jantung dan pencegahan berdasarkan pola gaya hidup yang dirancang untuk membantu masyarakat mengenali tingkat risiko penyakit jantung sejak dini. Di tengah meningkatnya kasus penyakit jantung sebagai salah satu penyebab kematian tertinggi di dunia, kesadaran masyarakat terhadap pentingnya menjaga kesehatan jantung melalui pola hidup sehat menjadi hal yang sangat diperlukan. Faktor-faktor seperti kebiasaan merokok, kurangnya aktivitas fisik, pola makan yang tidak sehat, stres, serta kualitas tidur yang buruk dapat meningkatkan risiko terjadinya penyakit jantung apabila tidak dikendalikan sejak dini.

Heartify hadir sebagai media skrining mandiri yang memberikan analisis risiko berdasarkan data profil pengguna dan kebiasaan gaya hidup sehari-hari. Aplikasi ini tidak hanya menampilkan tingkat risiko penyakit jantung, tetapi juga memberikan informasi mengenai faktor risiko utama yang paling berpengaruh serta rekomendasi pencegahan yang dapat diterapkan oleh pengguna, seperti pengaturan pola makan, peningkatan aktivitas fisik, pengelolaan stres, dan rekomendasi olahraga sesuai kondisi pengguna.

Heartify mampu memberikan analisis risiko secara cepat, mudah, dan personal. Selain itu, aplikasi ini juga menyediakan fitur penyimpanan riwayat analisis sehingga pengguna dapat memantau perubahan kondisi kesehatannya dari waktu ke waktu. Melalui pendekatan preventif ini, *Heartify* diharapkan dapat membantu masyarakat meningkatkan kesadaran akan pentingnya menjaga kesehatan jantung serta mendorong penerapan gaya hidup sehat untuk mengurangi risiko penyakit jantung di masa mendatang.

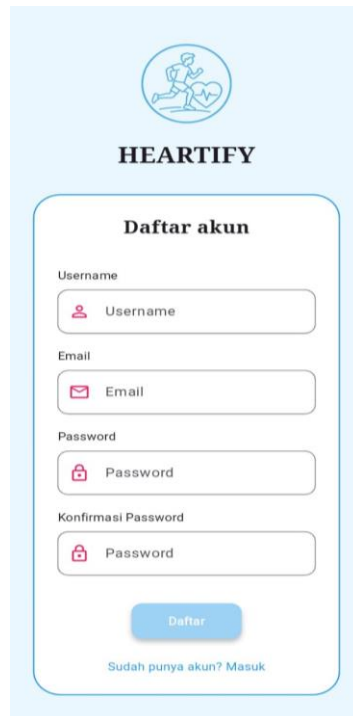
PENGUNAAN APLIKASI

1. REGISTER

Fitur Registrasi merupakan fitur yang digunakan oleh pengguna yang belum memiliki akun untuk melakukan pendaftaran ke dalam aplikasi *Heartify*. Proses registrasi menjadi langkah awal agar pengguna dapat memanfaatkan seluruh layanan yang tersedia pada aplikasi. Setiap akun yang dibuat akan tersimpan pada basis data sehingga dapat digunakan kembali saat pengguna ingin masuk ke aplikasi.

Fitur Registrasi memiliki beberapa fungsi sebagai berikut:

- Memungkinkan pengguna membuat akun baru pada aplikasi *Heartify*.
- Menyimpan data identitas awal pengguna, seperti username, alamat email, dan password.
- Mencegah penggunaan aplikasi oleh pengguna yang tidak terdaftar.
- Menjadi identitas pengguna sehingga hasil analisis dapat disimpan secara pribadi.
- Menjadi dasar proses autentikasi sebelum pengguna dapat mengakses fitur lainnya.



Gambar 2. 34 Halaman Register

2. LOGIN

Fitur Login digunakan untuk mengautentikasi pengguna agar dapat mengakses seluruh fitur pada aplikasi *Heartify*. Pengguna dapat masuk menggunakan **email dan password** yang telah terdaftar atau menggunakan **akun Google** untuk proses login yang lebih cepat dan praktis. Setelah berhasil login, pengguna dapat mengakses fitur profil, analisis risiko penyakit jantung, serta riwayat hasil analisis. Fitur Verifikasi LOGIN berfungsi untuk:

- Memverifikasi identitas pengguna.
- Menghindari pembuatan akun menggunakan email yang tidak aktif.
- Menjaga keamanan data pengguna agar tidak dapat diakses oleh pihak lain.
- Menghubungkan data profil dan riwayat analisis dengan akun pengguna.

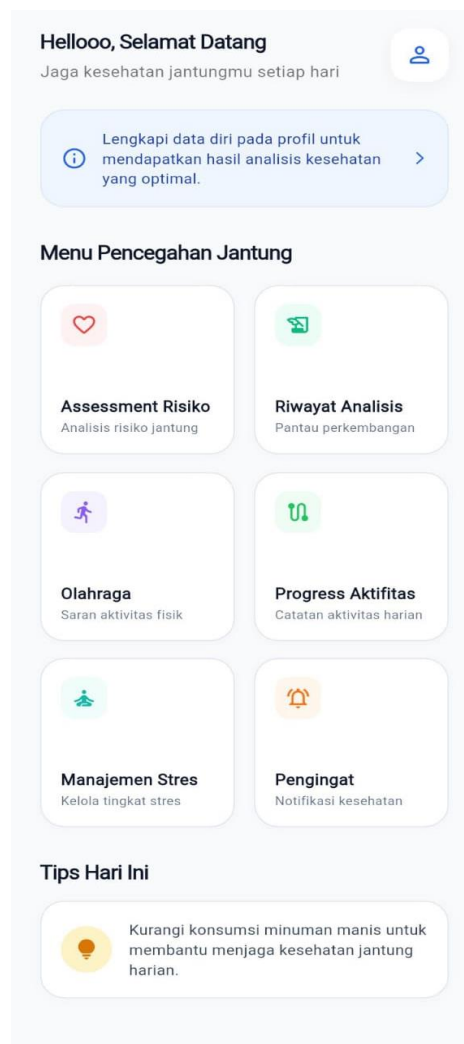


Gambar 2. 35 Halaman Login

3. HOME

Fitur **Home** merupakan halaman utama yang ditampilkan setelah pengguna berhasil login. Halaman ini berfungsi sebagai pusat navigasi untuk mengakses

seluruh fitur yang tersedia pada aplikasi *Heartify* . Melalui halaman Home, pengguna dapat dengan mudah memilih menu seperti Profil, Analisis Risiko Jantung, Rekomendasi Gaya Hidup Sehat, dan Riwayat Analisis. Tampilan Home dirancang sederhana dan mudah dipahami sehingga memudahkan pengguna dalam menggunakan aplikasi.



Gambar 2. 36 Halaman Home

4. PROFILE

Fitur Profil digunakan untuk menyimpan informasi pribadi pengguna yang akan digunakan sebagai data dasar dalam proses analisis risiko penyakit jantung.

Informasi yang dimasukkan meliputi nama lengkap, tanggal lahir, jenis kelamin, tinggi badan, dan berat badan.

Selain itu, sistem akan menghitung nilai Body Mass Index (BMI) secara otomatis berdasarkan tinggi badan dan berat badan yang dimasukkan oleh pengguna.

Fitur Profil memiliki beberapa fungsi, yaitu:

- Menyimpan informasi identitas pengguna.
- Menghitung usia secara otomatis berdasarkan tanggal lahir.
- Menghitung nilai BMI secara otomatis.
- Menampilkan kategori BMI (Kurus, Normal, Gemuk, atau Obesitas).
- Menjadi data dasar dalam proses analisis risiko penyakit jantung.
- Memungkinkan pengguna memperbarui data apabila terjadi perubahan berat badan maupun tinggi badan.

The screenshot shows a mobile application interface for a user profile. At the top, there is a back arrow and the title 'Profil Pengguna'. Below the title is a circular placeholder for a profile picture. The form contains the following fields:

- Nama Lengkap**: wafah lailatul
- 1/1/2000** (Date of Birth)
- Jenis Kelamin**: Perempuan
- Tinggi Badan (cm)**: 170.0
- Berat Badan (kg)**: 85.0

Below the form, a blue box displays the **Body Mass Index** as **29.4**, with the label **Gemuk** underneath. At the bottom, there are two buttons: a blue 'Simpan Profil' button and a red-outlined 'Logout' button with an arrow icon.

Gambar 2. 37 Halaman Profile

5. ANALISIS RESIKO JANTUNG

Fitur Analisis Risiko merupakan fitur utama pada aplikasi *Heartify*. Pengguna diminta mengisi beberapa parameter gaya hidup yang berhubungan dengan risiko penyakit jantung, seperti kebiasaan merokok, lingkungan perokok, aktivitas fisik, durasi tidur, tingkat stres, dan pola makan.

Setelah seluruh data diisi, sistem akan melakukan perhitungan skor risiko berdasarkan algoritma yang telah diterapkan pada aplikasi.

Fitur Analisis Risiko memiliki fungsi sebagai berikut:

- Mengumpulkan data gaya hidup pengguna.
- Mengidentifikasi faktor-faktor yang dapat meningkatkan risiko penyakit jantung.
- Menghitung skor risiko penyakit jantung.
- Menentukan kategori risiko pengguna.
- Mengidentifikasi faktor risiko yang paling dominan.
- Memberikan dasar dalam penyusunan rekomendasi pencegahan.

← Analisis Risiko Jantung

29.4
BMI: Gemuk

26 Tahun
Umur

Perempuan
Jenis Kelamin

Umur, jenis kelamin, dan BMI diambil otomatis dari profil Anda.

Gaya Hidup & Kebiasaan

Status Merokok
Tidak Merokok

Sering di lingkungan perokok?
Tidak

Aktivitas (menit/minggu)

Tidur (jam/hari)

Tingkat Stres
Rendah

Pilih tingkat stres sesuai dengan kondisi yang Anda rasakan.

Pola Makan
Sehat

Pilih kategori yang paling sesuai dengan kebiasaan makan Anda sehari-hari.

Analisis Risiko Sekarang

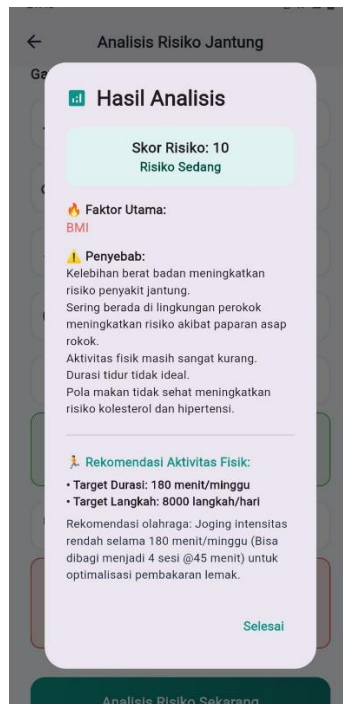
Gambar 2. 38 Halaman Analisis

6. HASIL ANALISIS

Setelah proses analisis selesai dilakukan, sistem akan menampilkan hasil analisis yang berisi informasi mengenai tingkat risiko penyakit jantung pengguna. Informasi tersebut disajikan secara ringkas sehingga mudah dipahami.

Fitur Hasil Analisis berfungsi untuk:

- Menampilkan skor risiko penyakit jantung.
- Menampilkan kategori risiko (Rendah, Sedang, atau Tinggi).
- Menampilkan faktor utama yang memengaruhi hasil analisis.
- Menjelaskan penyebab meningkatnya risiko penyakit jantung.
- Memberikan rekomendasi aktivitas fisik yang sesuai.
- Menampilkan target durasi olahraga setiap minggu.
- Menampilkan target langkah harian.
- Memberikan saran pola hidup sehat sebagai upaya pencegahan penyakit jantung.



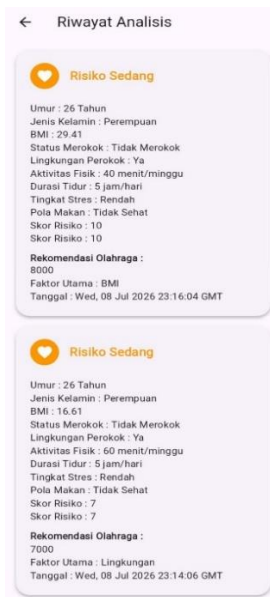
Gambar 2. 39 Halaman Hasil Analisis

7. RIWAYAT ANALISIS

Fitur Riwayat Analisis digunakan untuk menyimpan seluruh hasil analisis yang telah dilakukan oleh pengguna. Setiap hasil analisis akan disimpan secara otomatis sehingga dapat diakses kembali sewaktu-waktu.

Fitur Hasil Analisis berfungsi untuk:

- Fitur Riwayat Analisis memiliki fungsi sebagai berikut:
- Menyimpan seluruh hasil analisis pengguna.
- Menampilkan riwayat analisis berdasarkan waktu pelaksanaan.
- Membantu pengguna memantau perkembangan kondisi kesehatannya.
- Menjadi bahan evaluasi terhadap perubahan gaya hidup yang telah dilakukan.
- Memudahkan pengguna membandingkan hasil analisis sebelumnya dengan hasil analisis terbaru.



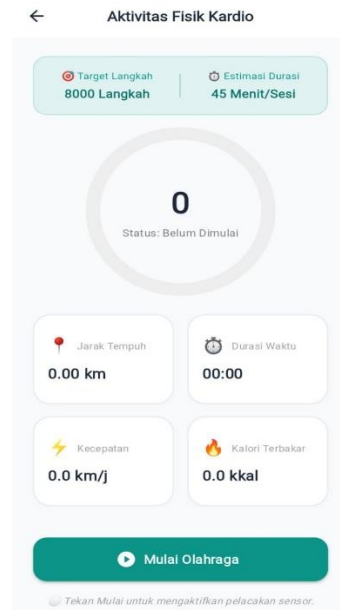
Gambar 2. 40 Riwayat Analisis

8. AKTIVITAS OLAHRAGA

Fitur **Aktivitas Olahraga** digunakan untuk memantau aktivitas jogging pengguna secara real-time. Sistem menampilkan informasi berupa *jUMLah* langkah, jarak tempuh, kecepatan, dan durasi olahraga. Target olahraga diberikan berdasarkan hasil analisis risiko yang mempertimbangkan umur, BMI, dan tingkat aktivitas fisik pengguna.

Fitur Hasil Analisis berfungsi untuk:

- Mendeteksi aktivitas jogging pengguna.
- Menghitung *jUMLah* langkah yang telah ditempuh.
- Menghitung jarak tempuh selama berolahraga.
- Menampilkan kecepatan dan durasi olahraga.
- Memberikan target olahraga yang disesuaikan dengan hasil analisis risiko penyakit jantung.
- Membantu pengguna mencapai target aktivitas fisik untuk menjaga kesehatan jantung.



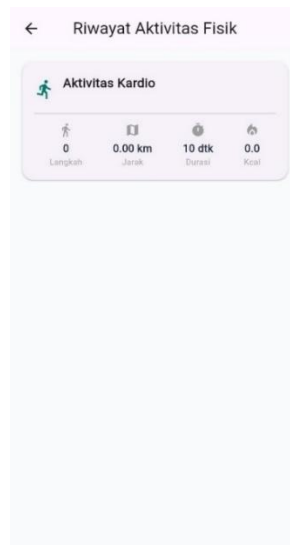
Gambar 2. 41 Aktivitas Olahraga

9. RIWAYAT OLAHRAGA

Fitur Riwayat Olahraga digunakan untuk menyimpan seluruh data aktivitas olahraga yang telah dilakukan oleh pengguna sehingga dapat dipantau kembali kapan saja.

Fitur Hasil Analisis berfungsi untuk:

- Menyimpan riwayat aktivitas olahraga.
- Menampilkan data langkah, jarak, durasi, dan waktu olahraga.
- Membantu pengguna memantau perkembangan aktivitas fisik.
- Menjadi bahan evaluasi terhadap konsistensi olahraga pengguna.



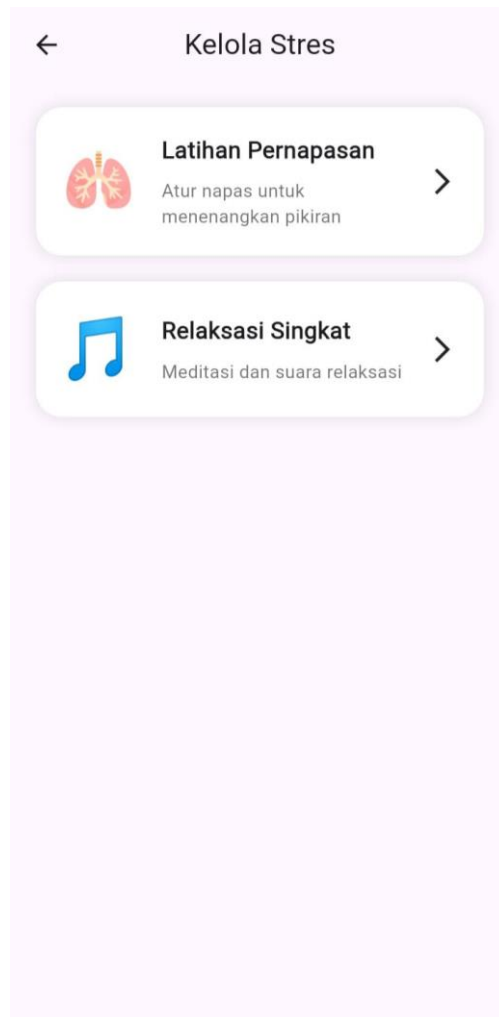
Gambar 2. 42 Halaman Riwayat Olahraga

10. MANEJEMEN STRESS

Fitur Pemantauan Stres menyediakan berbagai metode relaksasi untuk membantu pengguna mengurangi tingkat stres. Pengguna dapat memilih jenis relaksasi seperti latihan pernapasan maupun mendengarkan musik relaksasi.

Fitur Hasil Analisis berfungsi untuk:

- Membantu pengguna mengelola tingkat stres.
- Menyediakan pilihan latihan pernapasan.
- Menyediakan musik relaksasi untuk membantu menenangkan pikiran.
- Mendukung penerapan gaya hidup sehat sebagai upaya pencegahan penyakit jantung.



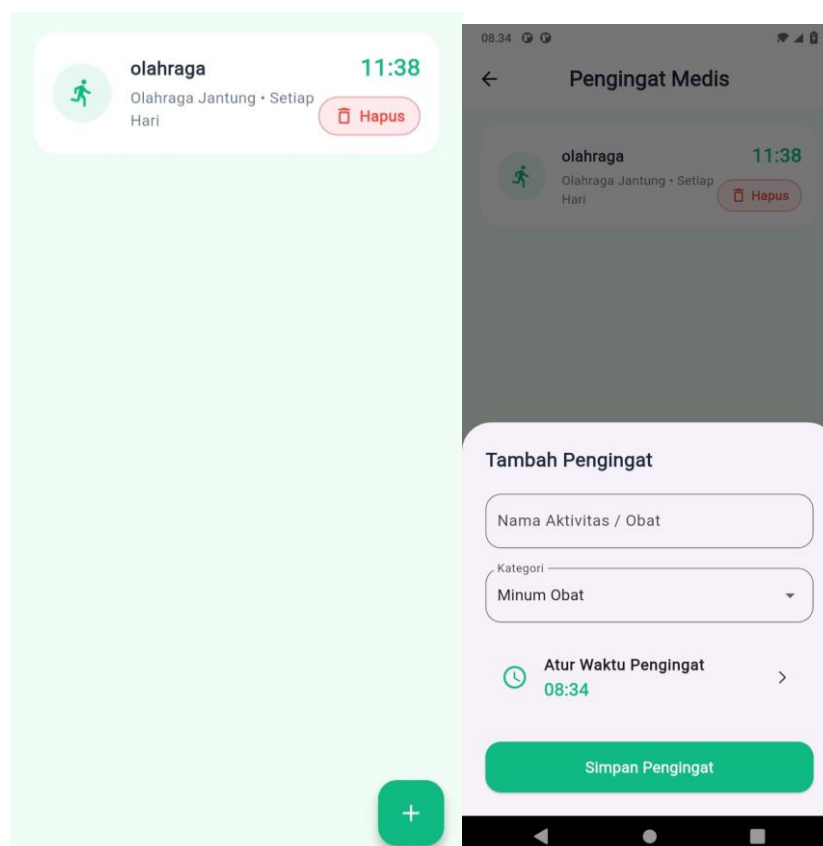
Gambar 2. 43 Halaman Manajemen Stress

11. FITUR PENGINGAT

Fitur Pengingat digunakan untuk membantu pengguna menjalankan pola hidup sehat secara teratur melalui pemberian notifikasi sesuai jadwal yang telah ditentukan.

Fitur Hasil Analisis berfungsi untuk:

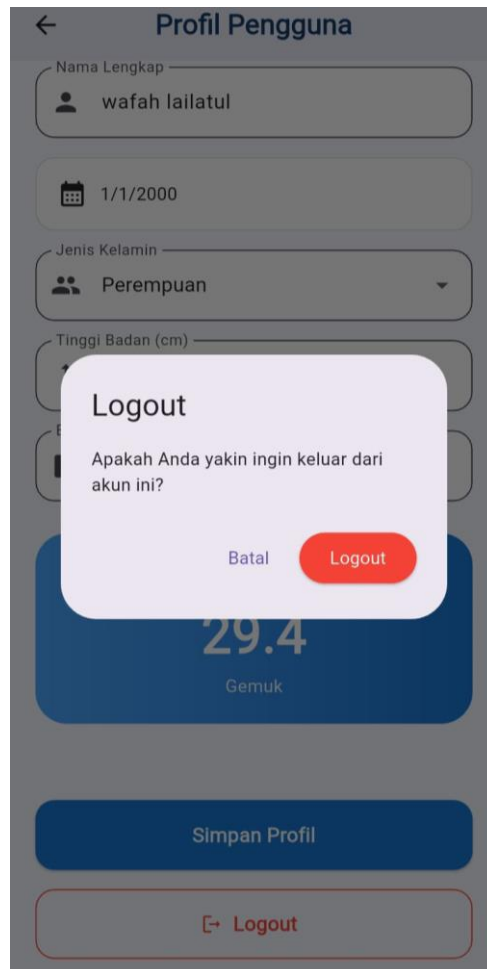
- Mengingatkan jadwal minum obat.
- Mengingatkan jadwal olahraga.
- Mengingatkan waktu tidur.
- Mengingatkan minum air putih atau aktivitas sehat lainnya.
- Membantu meningkatkan kedisiplinan pengguna dalam menerapkan gaya hidup sehat.



Gambar 2. 44 Halaman Pengingat

12. LOGOUT

Fitur Logout digunakan untuk mengakhiri sesi penggunaan aplikasi dan keluar dari akun pengguna. Setelah proses logout dilakukan, pengguna harus melakukan login kembali untuk dapat mengakses aplikasi.



Gambar 2. 45 Halaman Logout

HEARTIFY

Your Heart, Your Healthy
Future



DUKOMEN TEKNIKAL

WAFAH LAILATUL ISTIANAH

DWI INTAN AF'IDAH, S.T., M.KOM

ARIF HIDAYAH, S.TR.KOM., M.TR.KOM

DAFTAR ISI

DAFTAR ISI	18
1. Pendahuluan.....	19
1.1. Latar Belakang	19
1.2. Batasan Masalah.....	21
1.3 Tujuan	22
2. Spesifikasi Teknis dan Lingkungan Pengembangan.....	23
2.1 Spesifikasi Teknis.....	23
2.2 Lingkungan Pengembangan	24
3. ARSITEKTUR PERANGKAT LUNAK.....	24
3.1 Gambaran Arsitektur Perangkat Lunak	24
3.2 Struktur Perangkat Lunak	25
3.2.1 Struktur <i>Frontend</i>	25
3.2.2. Struktur <i>Backend</i>	27
4. Implementasi Sistem	29
4.1. Implementasi Koneksi <i>MongoDB</i>	29
4.2 Implementasi Register	30
4.3 Implementasi Verivity OTP	31
4.4 Implementasi Login.....	33
4.5 Implementasi Profile.....	33
4.6 Implementasi Analisis Risiko Penyakit Jantung	34
4.7 Implementasi Riwayat Analisis.....	37
4.8 Implementasi Aktivitas Fisik	40
4.9 Implentasi Fitur Pengingat	40

1. Pendahuluan

1.1. Latar Belakang

Meningkatnya kasus penyakit jantung berkaitan erat dengan perubahan gaya hidup masyarakat modern. Penyakit jantung masih menjadi salah satu penyebab kematian tertinggi di dunia dan merupakan tantangan besar dalam bidang kesehatan. Penyakit kardiovaskular tetap menjadi penyumbang angka kematian tertinggi secara global. Kondisi ini menunjukkan bahwa meskipun perkembangan teknologi medis semakin maju, penyakit jantung masih menjadi ancaman serius yang memerlukan perhatian lebih [1].

Saat ini, penyakit jantung tidak hanya dialami oleh kelompok usia lanjut, tetapi *API* juga mulai banyak ditemukan pada usia produktif. Peningkatan kasus penyakit kardiovaskular pada kelompok usia muda. Kondisi ini menjadi perhatian karena dapat menurunkan kualitas hidup masyarakat sekaligus berdampak pada produktivitas dan beban pembiayaan kesehatan [2].

Kurangnya aktivitas fisik, kebiasaan duduk dalam waktu yang lama, pola makan yang tidak sehat, merokok, konsumsi alkohol, serta stres menjadi faktor-faktor yang berkontribusi terhadap meningkatnya risiko penyakit jantung. Hasil penelitian *Scientific Reports* menunjukkan bahwa kombinasi faktor-faktor tersebut dapat mempercepat kerusakan pembuluh darah dan meningkatkan risiko terjadinya penyakit kardiovaskular [3].

Gaya hidup yang tidak sehat dalam jangka panjang dapat memicu berbagai faktor risiko, seperti obesitas, hipertensi, dislipidemia, dan peningkatan kadar

kolesterol. kondisi tersebut berkembang secara bertahap dan sering kali tidak menimbulkan gejala pada tahap awal. Oleh karena itu, deteksi dini terhadap faktor-faktor risiko menjadi langkah penting untuk mencegah terjadinya penyakit jantung yang lebih serius [4].

Seiring berkembangnya pendekatan *preventif* dalam dunia kesehatan, analisis risiko sejak dini menjadi salah satu strategi yang dinilai efektif untuk menurunkan angka kejadian penyakit jantung. pemetaan tingkat risiko berdasarkan karakteristik individu dan gaya hidup dapat membantu mengidentifikasi potensi penyakit sebelum muncul gejala klinis [5].

Salah satu pendekatan yang dapat digunakan adalah Machine Learning, yaitu metode yang memungkinkan sistem mempelajari pola dari data untuk menghasilkan prediksi. Dalam penelitian ini, algoritma *Random Forest* digunakan untuk melakukan *Klasifikasi* tingkat risiko penyakit jantung berdasarkan karakteristik pengguna dan faktor gaya hidup. *Random Forest* merupakan algoritma ensemble learning yang menggabungkan sejumlah decision tree untuk menghasilkan prediksi melalui proses penggabungan hasil dari beberapa pohon keputusan [6]. Penerapan algoritma *Random Forest* dalam prediksi penyakit jantung juga telah dilakukan pada penelitian terdahulu, yang menunjukkan bahwa algoritma tersebut dapat digunakan untuk melakukan *Klasifikasi* berdasarkan data pasien [7].

Pentingnya upaya pencegahan juga didukung oleh penelitian yang menyatakan bahwa sekitar 80% kasus penyakit jantung koroner dapat dicegah melalui perubahan gaya hidup yang sehat [2]. Hal ini menunjukkan bahwa

penerapan pola hidup sehat memiliki peran yang sangat besar dalam mengurangi risiko terjadinya penyakit jantung.

Kemajuan teknologi digital memberikan peluang dalam pengembangan aplikasi kesehatan untuk mendukung deteksi dini dan pencegahan penyakit. Melalui perangkat seluler, pengguna dapat mengakses informasi kesehatan dan melakukan pemantauan terhadap faktor-faktor yang berkaitan dengan kondisi kesehatannya. Pemanfaatan teknologi tersebut dapat menjadi salah satu sarana untuk membantu pengguna memahami kondisi kesehatan serta menerapkan pola hidup yang lebih sehat [8].

Berdasarkan uraian tersebut, diperlukan suatu aplikasi yang mampu menganalisis risiko penyakit jantung berdasarkan faktor gaya hidup serta memberikan rekomendasi pencegahan yang sesuai dengan kondisi pengguna. Oleh karena itu, penelitian ini mengembangkan aplikasi *Heartify* yang menerapkan metode Machine Learning dengan algoritma *Random Forest* untuk melakukan *Klasifikasi* tingkat risiko penyakit jantung berdasarkan karakteristik pengguna dan faktor gaya hidup. Hasil *Klasifikasi* tersebut digunakan sebagai dasar dalam memberikan informasi dan rekomendasi pencegahan yang sesuai dengan kondisi pengguna. Sistem ini diharapkan dapat membantu masyarakat mengenali tingkat risiko penyakit jantung sejak dini serta menjadi media edukasi dalam menerapkan pola hidup sehat sebagai upaya pencegahan penyakit jantung.

1.2. Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan membangun aplikasi *Heartify* berbasis mobile yang dapat melakukan analisis risiko penyakit jantung berdasarkan pola gaya hidup pengguna?
2. Bagaimana menerapkan metode Machine Learning dengan algoritma *Random Forest* dalam melakukan *Klasifikasi* tingkat risiko penyakit jantung berdasarkan karakteristik pengguna dan faktor-faktor gaya hidup?
3. Bagaimana memberikan rekomendasi pencegahan yang sesuai dengan tingkat risiko dan faktor risiko utama pengguna?
4. Bagaimana mengetahui tingkat kelayakan dan usability aplikasi *Heartify* dalam membantu pengguna melakukan analisis risiko dan pencegahan penyakit jantung?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Merancang dan membangun aplikasi *Heartify* berbasis mobile untuk melakukan penilaian risiko gaya hidup sebagai upaya pencegahan penyakit jantung.
2. Menerapkan algoritma *Random Forest* untuk mempelajari pola dari data faktor risiko penyakit jantung dan menghasilkan prediksi tingkat risiko pengguna.
3. Menyajikan rekomendasi pencegahan yang dipersonalisasi (personalized) serta pengingat (reminder) berdasarkan Faktor Risiko Utama (Top Factor) gaya hidup masing-masing pengguna.
4. Melakukan pengujian sistem (uji kelayakan dan usability) kepada pengguna

untuk memastikan aplikasi berjalan dengan baik, ramah pengguna, dan efektif digunakan dalam memantau kesehatan.

2. Spesifikasi Teknis dan Lingkungan Pengembangan

2.1 Spesifikasi Teknis

Spesifikasi teknis menjelaskan perangkat keras dan perangkat lunak yang digunakan dalam proses pengembangan aplikasi *Heartify*. Aplikasi dikembangkan menggunakan *framework Flutter* sebagai *frontend*, *Flask* sebagai *backend*, serta *MongoDB Atlas* sebagai basis data. Komunikasi antara aplikasi dan *server* dilakukan melalui *REST API* sehingga proses pertukaran data dapat berjalan dengan cepat dan terintegrasi.

a. Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Sistem Operasi	Windows 11 64-bit
Processor	Intel Core i5 atau setara
RAM	8 GB
Penyimpanan	SSD 256 GB
Smartphone	<i>Android</i> minimal versi 8.0 (Oreo)
Koneksi Internet	Digunakan untuk mengakses <i>API</i> dan <i>MongoDB Atlas</i>

b. Spesifikasi Perangkat Lunak

Perangkat Lunak	Fungsi
<i>Flutter SDK</i>	Pengembangan aplikasi mobile
Dart	Bahasa pemrograman <i>frontend</i>
Python 3	Bahasa pemrograman <i>backend</i>
<i>Flask</i>	<i>Framework backend REST API</i>
<i>MongoDB Atlas</i>	Basis data NoSQL
Visual Studio Code	Editor kode
Git	Version Control
Postman	Pengujian <i>REST API</i>

2.2 Lingkungan Pengembangan

Lingkungan pengembangan merupakan kombinasi perangkat keras dan perangkat lunak yang digunakan selama proses pembangunan aplikasi **Heartify** . Pengembangan dilakukan menggunakan sistem operasi **Windows 11** dengan editor **Visual Studio Code**. *Framework Flutter* digunakan untuk membangun antarmuka aplikasi mobile berbasis *Android*, sedangkan **Flask** digunakan untuk mengembangkan layanan *backend* yang bertugas mengelola proses autentikasi pengguna, pengolahan data profil, pengolahan data masukan, serta proses *Klasifikasi* risiko penyakit jantung menggunakan metode **Random Forest**.

Basis data yang digunakan adalah **MongoDB Atlas**, yang berfungsi untuk menyimpan data pengguna, profil, riwayat analisis, aktivitas fisik, pengingat, dan informasi lainnya. Model **Random Forest** digunakan untuk melakukan *Klasifikasi* tingkat risiko berdasarkan data faktor risiko yang telah diproses sebelumnya. Komunikasi antara aplikasi mobile dan *backend* dilakukan menggunakan **REST API** dengan format pertukaran data **JSON**.

Selama proses pengembangan dan pengujian, aplikasi dijalankan menggunakan emulator *Android* maupun perangkat *Android* secara langsung, sedangkan pengujian *endpoint backend* dilakukan menggunakan **Postman**.

3. ARSITEKTUR PERANGKAT LUNAK

3.1 Gambaran Arsitektur Perangkat Lunak

Arsitektur perangkat lunak merupakan rancangan yang menggambarkan struktur

serta hubungan antar komponen dalam aplikasi *Heartify* . Arsitektur ini digunakan sebagai acuan dalam proses pengembangan agar setiap komponen dapat bekerja secara terintegrasi dan mendukung seluruh fungsionalitas aplikasi.

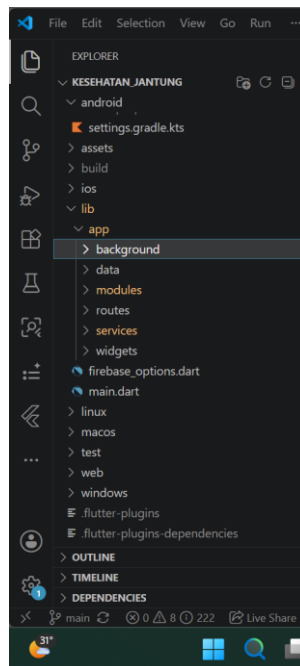
Heartify menerapkan arsitektur *client-server* , di mana aplikasi mobile yang dibangun menggunakan *Flutter* berperan sebagai client, sedangkan *Flask* berperan sebagai *server* yang menyediakan layanan *REST API*. Seluruh data pengguna disimpan pada *MongoDB Atlas* sebagai basis data NoSQL. Komunikasi antara aplikasi dan *server* menggunakan protokol HTTP dengan format pertukaran data *JSON*. Melalui arsitektur ini, proses autentikasi, analisis risiko penyakit jantung, pengelolaan aktivitas fisik, pengingat kesehatan, manajemen stres, serta riwayat analisis dapat dilakukan secara terpusat.

3.2 Struktur Perangkat Lunak

Struktur perangkat lunak menjelaskan organisasi file dan folder yang digunakan dalam pengembangan aplikasi *Heartify* . Penyusunan struktur ini bertujuan untuk memisahkan setiap komponen berdasarkan fungsinya sehingga proses pengembangan, pemeliharaan, dan pengembangan fitur baru dapat dilakukan dengan lebih mudah. Secara umum, struktur perangkat lunak *Heartify* dibagi menjadi dua bagian utama, yaitu *frontend* dan *backend* .

3.2.1 Struktur *Frontend*

Frontend dikembangkan menggunakan *framework Flutter* dengan bahasa pemrograman Dart. Struktur folder pada *frontend* dirancang berdasarkan fungsi masing-masing komponen agar kode lebih terorganisir dan mudah dikelola.



Keterangan struktur *frontend*:

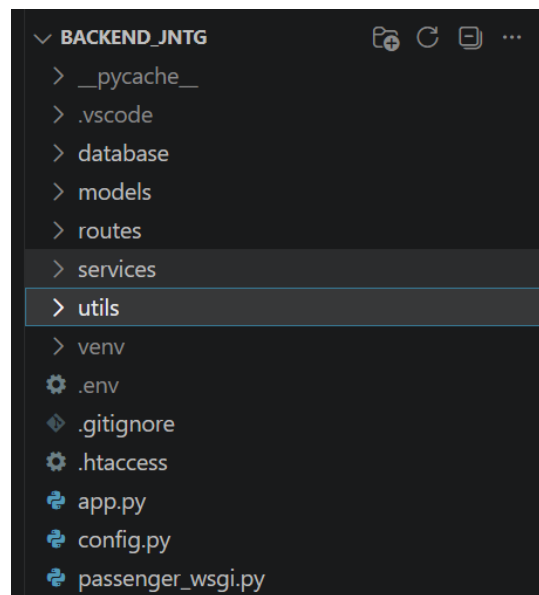
- **Models** digunakan untuk mendefinisikan model atau struktur data yang digunakan pada aplikasi, seperti data pengguna, hasil analisis, aktivitas fisik, dan pengingat.
- **Services** berisi kelas yang menangani komunikasi antara aplikasi dengan *backend* melalui *REST API*, seperti proses login, registrasi, penyimpanan profil, dan pengambilan data.
- **Views** berisi seluruh halaman antarmuka pengguna, seperti halaman login, registrasi, beranda, profil, analisis risiko, riwayat analisis, aktivitas fisik, pengingat, dan manajemen stres.
- **Widgets** berisi komponen antarmuka yang dapat digunakan kembali pada beberapa halaman, seperti tombol, kartu informasi, dialog, dan komponen lainnya.
- **Firestore_options.dart** berisi konfigurasi *Firestore* yang digunakan oleh aplikasi *Heartify*. File ini dibuat secara otomatis saat proses integrasi *Firestore* dan digunakan untuk menghubungkan aplikasi dengan layanan *Firestore* sesuai platform yang digunakan.
- **Routes** digunakan untuk mengatur proses perpindahan halaman (navigation) antar fitur pada aplikasi. Dengan adanya pengaturan *routing*, setiap halaman dapat diakses melalui rute yang telah ditentukan sehingga navigasi aplikasi menjadi

lebih terstruktur.

- **Modules** merupakan bagian utama yang berisi implementasi setiap fitur pada aplikasi *Heartify* . Setiap fitur dikelompokkan ke dalam modul tersendiri sehingga kode program menjadi lebih terstruktur dan mudah dipelihara.
- **Main.dart** merupakan file utama yang berfungsi sebagai titik awal untuk menjalankan aplikasi *Heartify* .

3.2.2. Struktur *Backend*

Backend dikembangkan menggunakan *framework Flask* dengan bahasa pemrograman **Python**. *Backend* bertugas mengelola logika bisnis aplikasi, autentikasi pengguna, proses analisis risiko penyakit jantung, serta komunikasi dengan basis data *MongoDB Atlas*.



Keterangan struktur *backend* :

- **app.py** merupakan file utama yang digunakan untuk menjalankan aplikasi *backend* dan menginisialisasi seluruh konfigurasi sistem.
- **routes/** berisi kumpulan endpoint *REST API* yang digunakan untuk menangani permintaan dari aplikasi Flutter, seperti login, registrasi, penyimpanan profil, analisis risiko, aktivitas fisik, pengingat, dan riwayat analisis.
- **database/** berisi konfigurasi koneksi ke *MongoDB Atlas* serta pengelolaan akses

terhadap basis data.

- **middleware/** digunakan untuk menangani proses autentikasi dan validasi token JWT sebelum pengguna mengakses layanan tertentu.
- **utils/** berisi fungsi-fungsi pendukung yang digunakan oleh *backend* , seperti validasi data atau fungsi utilitas lainnya.
- **requirements.txt** berisi daftar pustaka (library) Python yang diperlukan agar *backend* dapat berjalan dengan baik.

3.2.3 Struktur Basis Data

Struktur basis data pada aplikasi **Heartify** menggunakan **MongoDB Atlas** sebagai sistem manajemen basis data berbasis **NoSQL**. Data disimpan dalam bentuk dokumen (*document*) dengan format **BSON**, sehingga memberikan fleksibilitas dalam penyimpanan dan pengelolaan data sesuai kebutuhan aplikasi.

Basis data dirancang untuk mendukung seluruh fitur utama aplikasi, seperti autentikasi pengguna, pengelolaan profil, analisis risiko penyakit jantung, riwayat analisis, aktivitas fisik, dan pengingat kesehatan. Setiap data disimpan pada *collection* sesuai fungsinya dan dihubungkan menggunakan *user_id* agar proses pengelolaan data menjadi lebih terstruktur. Rancangan struktur basis data aplikasi *Heartify* .

Collection name	Properties	Storage size	Data size	Documents	Avg. document size	Indexes	Total index size
aktivitas_fisik	-	36.86 kB	1.16 kB	10	116.00 B	1	36.86 kB
analysis_history	-	36.86 kB	4.05 kB	8	506.00 B	1	36.86 kB
otp_register	-	36.86 kB	313.00 B	1	313.00 B	1	36.86 kB
otp_reset_password	-	36.86 kB	302.00 B	3	100.00 B	1	36.86 kB
otp_verification	-	36.86 kB	132.00 B	1	132.00 B	1	36.86 kB
profiles	-	36.86 kB	689.00 B	3	229.00 B	1	36.86 kB
reminders	-	36.86 kB	660.00 B	4	165.00 B	1	36.86 kB
user_tokens	-	36.86 kB	14.11 kB	43	328.00 B	1	36.86 kB
users	-	36.86 kB	2.32 kB	7	331.00 B	1	36.86 kB

4. Implementasi Sistem

4.1. Implementasi Koneksi *MongoDB*

implementasi koneksi antara *backend Flask* dengan database *MongoDB Atlas*. Konfigurasi koneksi diawali dengan memuat variabel lingkungan menggunakan `load_dotenv()`, kemudian mengambil nilai `MONGO_URI` dari file `.env` agar informasi koneksi tidak dituliskan secara langsung pada source code. Selanjutnya, objek `MongoClient` digunakan untuk membangun koneksi ke *MongoDB Atlas* dengan dukungan sertifikat SSL melalui library `certifi` sehingga komunikasi data berlangsung secara aman. Setelah koneksi berhasil dibuat, aplikasi memilih database `kesehatan_jantung` sebagai basis data utama yang digunakan untuk menyimpan dan mengelola seluruh data pada aplikasi *Heartify*.

```

import os
from dotenv import load_dotenv
from pymongo import MongoClient
import certifi

load_dotenv()

MONGO_URI = os.getenv("MONGO_URI")

client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())

try:
    client.admin.command("ping")
    print("MongoDB Atlas Connected")

except Exception as e:
    print("MongoDB Connection Error:", e)

db = client["kesehatan_jantung"]

```

4.2 Implementasi Register

implementasi proses registrasi pengguna pada *backend Flask*. Sistem menerima data berupa username, email, dan password dari aplikasi *Flutter* melalui metode HTTP POST. Selanjutnya dilakukan validasi untuk memastikan username atau email belum terdaftar pada database. Jika data valid, password akan diamankan menggunakan algoritma *bcrypt* sebelum disimpan ke collection *users* di *MongoDB Atlas*. Setelah proses penyimpanan berhasil, *backend* mengirimkan respons dalam format *JSON* sebagai tanda bahwa registrasi berhasil dilakukan.

```

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

    username = data.get('username', '').strip()
    email = data.get('email', '').strip().lower()
    password = generate_password_hash(data.get('password', '').strip())

    users = db["users"]
    otp_register = db["otp_register"]

    # cek email sudah terdaftar
    if users.find_one({"email": email}):
        return jsonify({
            "status": "error",
            "message": "Email sudah terdaftar"
        }), 400

    # generate OTP
    otp = str(random.randint(100000, 999999))

    # hapus OTP lama jika ada
    otp_register.delete_many({
        "email": email
    })

```

```

# simpan OTP
otp_register.insert_one({
    "email": email,
    "username": username,
    "password": password,
    "otp": otp,
    "created_at": datetime.utcnow()
})

print("OTP REGISTER:", otp)

# kirim email
msg = Message(
    'Kode OTP Register',
    sender=app.config['MAIL_USERNAME'],
    recipients=[email]
)

msg.body = f'Kode OTP Anda: {otp}'

mail.send(msg)

return jsonify({
    "status": "success",
    "message": "OTP berhasil dikirim"
}), 200

```

4.3 Implementasi Verivity OTP

implementasi proses verifikasi OTP pada *backend Flask*. Sistem menerima email dan kode OTP dari aplikasi Flutter, kemudian mencocokkannya dengan data yang tersimpan pada collection `otp_register`. Jika OTP valid, data pengguna

dipindahkan ke collection users sebagai akun yang telah terverifikasi, kemudian data OTP dihapus dari database. Sebaliknya, apabila OTP tidak sesuai, sistem mengembalikan pesan bahwa kode OTP tidak valid. Setelah proses verifikasi berhasil, *backend* mengirimkan respons keberhasilan kepada aplikasi.

```
@app.route('/verify-otp-register', methods=['POST'])
def verify_otp_register():
    data = request.get_json()

    email = data.get("email", "").strip().lower()
    otp = data.get("otp", "").strip()

    otp_collection = db["otp_register"]
    users = db["users"]

    otp_data = otp_collection.find_one({
        "email": email,
        "otp": otp
    })

    print("EMAIL OTP:", email)
    print("OTP INPUT:", otp)
    print("OTP DATA:", otp_data)

    if not otp_data:
        return jsonify({
            "status": "error",
            "message": "OTP tidak valid"
        }), 400

    users.insert_one({
        "username": otp_data["username"],
        "email": otp_data["email"],
        "password": otp_data["password"],
        "created_at": datetime.utcnow()
    })

    otp_collection.delete_one({
        "_id": otp_data["_id"]
    })

    return jsonify({
        "status": "success",
        "message": "Verifikasi berhasil"
    }), 200
```

4.4 Implementasi Login

implementasi proses login pengguna pada *backend Flask*. Sistem menerima email dan password yang dikirim dari aplikasi Flutter, kemudian melakukan validasi dengan data yang tersimpan pada collection users. Jika data sesuai, sistem akan membuat *JSON Web Token (JWT)* yang digunakan sebagai identitas pengguna untuk mengakses fitur yang memerlukan autentikasi. Setelah proses login berhasil, *backend* mengirimkan token dan informasi pengguna kepada aplikasi.

```
@app.route('/login', methods=['POST'])
def login():

    data = request.get_json()

    email = data.get(
        "email",
        ""
    ).strip().lower()

    password = data.get(
        "password",
        ""
    ).strip()

    users = db["users"]

    # generate JWT
    token = generate_token(user)

    # simpan token MongoDB
    user_tokens = db["user_tokens"]

    user_tokens.insert_one({
        "user_id":
            str(user["_id"]),

        "token":
            token,

        "created_at":
            datetime.utcnow(),

        "expired at":
```

4.5 Implementasi Profile

implementasi pengelolaan profil pengguna pada *backend Flask*. Sistem

menerima data profil, seperti nama lengkap, tanggal lahir, jenis kelamin, tinggi badan, dan berat badan, kemudian menyimpannya ke dalam database *MongoDB*. Pada proses ini, sistem juga menghitung Body Mass Index (BMI) berdasarkan tinggi dan berat badan pengguna sebelum data disimpan. Informasi profil tersebut selanjutnya digunakan sebagai salah satu parameter dalam proses analisis risiko penyakit jantung.

```
@app.route(
    '/save-profile',
    methods=['POST']
)
@token_required
def save_profile():
    try:
        user_id = request.user["user_id"]

        data = request.get_json()

        if not data:
            return jsonify({
                "status": "error",
                "message": "Data kosong"
            }), 400

        full_name = data.get("full_name")
        birth_date = data.get("birth_date")
        gender = data.get("gender")
```

4.6 Implementasi Analisis Risiko Penyakit Jantung

Implementasi metode Machine Learning menggunakan algoritma *Random Forest* pada fitur analisis risiko penyakit jantung. Sistem mengumpulkan berbagai data faktor risiko dari pengguna, seperti usia, jenis kelamin, BMI, kebiasaan merokok, paparan lingkungan perokok, durasi aktivitas fisik, waktu tidur, tingkat stres, dan

pola makan. Data kondisi tersebut kemudian diubah ke dalam format numerik (*encoding*) dan diproses langsung oleh model *Random Forest* yang telah dilatih sebelumnya. Alih-alih menjumlahkan skor secara kaku, model ini bekerja dengan menganalisis pola yang kompleks dari berbagai kombinasi faktor risiko tersebut berdasarkan kecerdasan buatan. Dari hasil analisis dan prediksi model, sistem secara otomatis melakukan *Klasifikasi* ke dalam tiga kategori, yaitu risiko rendah, sedang, dan tinggi

```
import joblib
import pandas as pd
import os
import warnings

# Suppress sklearn warnings if needed
warnings.filterwarnings("ignore", category=UserWarning)

# Dapatkan path model
BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
MODEL_PATH = os.path.join(BASE_DIR, 'models', 'heart_model (1).pkl')

# Load model (lazy loading)
_model = None

def get_model():
    global _model
    if _model is None:
        _model = joblib.load(MODEL_PATH)
    return _model

# Mapping berdasarkan alphabetical LabelEncoding dari
generate_dataset.py
JENIS_KELAMIN_MAP = {'Laki-laki': 0, 'Perempuan': 1}
MEROKOK_MAP = {'Perokok Aktif': 0, 'Perokok Pasif': 1, 'Tidak
Merokok': 2}
LINGKUNGAN_MAP = {'Tidak': 0, 'Ya': 1}
STRES_MAP = {'Rendah': 0, 'Sedang': 1, 'Tinggi': 2}
POLA_MAKAN_MAP = {'Cukup Sehat': 0, 'Sehat': 1, 'Tidak Sehat': 2}
```

```

RISIKO_MAP = {0: 'Risiko Rendah', 1: 'Risiko Sedang', 2: 'Risiko
Tinggi'}

def predict_heart_risk(data: dict) -> dict:
    model = get_model()

    # 1. Parsing and mapping
    umur = int(data.get('age', 0))
    jenis_kelamin = JENIS_KELAMIN_MAP.get(data.get('gender', ''), 0)
    bmi = float(data.get('bmi', 0.0))
    merokok = MEROKOK_MAP.get(data.get('smoking', ''), 2)
    lingkungan = LINGKUNGAN_MAP.get(data.get('smokingEnvironment',
    ''), 0)
    olahraga_menit = int(data.get('activity', 0))
    tidur_jam = int(data.get('sleep', 0))
    stres = STRES_MAP.get(data.get('stress', ''), 0)
    pola_makan = POLA_MAKAN_MAP.get(data.get('eating', ''), 0)

    # Feature ordering must match exactly:
    # ['Umur' 'Jenis_Kelamin' 'BMI' 'Merokok' 'Lingkungan_Perokok'
    #  'Olahraga_Menit' 'Tidur_Jam' 'Stres' 'Pola_Makan']

    input_df = pd.DataFrame([
        'Umur': umur,
        'Jenis_Kelamin': jenis_kelamin,
        'BMI': bmi,
        'Merokok': merokok,
        'Lingkungan_Perokok': lingkungan,
        'Olahraga_Menit': olahraga_menit,
        'Tidur_Jam': tidur_jam,
        'Stres': stres,
        'Pola_Makan': pola_makan
    ])

    # Predict
    pred = model.predict(input_df)[0]
    category = RISIKO_MAP.get(pred, 'Risiko Rendah')

    # Generate recommendations based on the new class
    # Karena kita menggunakan ML, kita tidak lagi mengandalkan
    `score`.
    # Kita berikan score buatan (hanya untuk kompatibilitas UI jika
    UI masih butuh, atau 0 saja)
    # Rekomendasi olahraga
    if category == 'Risiko Tinggi':

```

```

        sportsDuration = 200
        sportsTargetSteps = 10000
        sportsRecommendation = "Lakukan olahraga kardio ringan-
sedang secara bertahap dengan pantauan dokter."
        topFactor = "Faktor Risiko Terdeteksi Tinggi"
        score = 15
    elif category == 'Risiko Sedang':
        sportsDuration = 150
        sportsTargetSteps = 7000
        sportsRecommendation = "Lakukan olahraga jalan santai
minimal 30 menit sehari."
        topFactor = "Pola Hidup Perlu Diperbaiki"
        score = 8
    else:
        sportsDuration = 150
        sportsTargetSteps = 7000
        sportsRecommendation = "Pertahankan aktivitas fisik rutin
untuk menjaga kesehatan jantung."
        topFactor = "Gaya Hidup Sehat"
        score = 3

    return {
        "category": category,
        "score": score,
        "topFactor": topFactor,
        "sportsDuration": sportsDuration,
        "sportsTargetSteps": sportsTargetSteps,
        "sportsRecommendation": sportsRecommendation
    }

```

4.7 Implementasi Riwayat Analisis

Implementasi proses analisis risiko dan penyimpanannya ke dalam database. Data faktor risiko yang diinputkan oleh pengguna dikirimkan dari aplikasi ke *backend* melalui layanan API. Di sisi *backend*, data tersebut diproses dan dievaluasi menggunakan model *Machine Learning Random Forest* untuk memprediksi kategori risiko, menentukan faktor utama, serta menghasilkan rekomendasi aktivitas. Setelah prediksi selesai, seluruh hasil analisis beserta data

kondisi awal pengguna tersebut secara otomatis disimpan ke dalam database. Data ini dikelola secara terpusat sehingga pengguna dapat melihat kembali riwayat analisis kesehatannya kapan saja.

```
@app.route('/save-analysis', methods=['POST'])
@token_required
def save_analysis():

    try:

        print("===== SAVE ANALYSIS =====")

        # ambil user dari JWT

        user_id = request.user["user_id"]

        print("USER ID :", user_id)

        data = request.get_json()
        print("BODY :", data)

        if not data:
            return jsonify({
                "status": "error",
                "message": "Data kosong"
            }), 400

        from services.ml_service import predict_heart_risk

        # Predict using Random Forest Model
        ml_result = predict_heart_risk(data)

        analysis = db["analysis_history"]

        result = {
            "user_id": user_id,
            "age": data.get("age"),
            "gender": data.get("gender"),
            "bmi": data.get("bmi"),
```

```

        "smoking": data.get("smoking"),
        "smoking_environment": data.get("smokingEnvironment"),
        "activity": data.get("activity"),
        "sleep": data.get("sleep"),
        "stress": data.get("stress"),
        "eating": data.get("eating"),

        "score": ml_result["score"],
        "category": ml_result["category"],
        "top_factor": ml_result["topFactor"],

        # rekomendasi olahraga
        "sports_duration": ml_result["sportsDuration"],
        "sports_target_steps": ml_result["sportsTargetSteps"],
        "sports_recommendation":
ml_result["sportsRecommendation"],

        "created_at": datetime.utcnow()
    }

    analysis.insert_one(result)

    return jsonify({
        "status": "success",
        "message": "Analisis berhasil disimpan",
        "data": ml_result
    }), 200

except Exception as e:

    import traceback

    traceback.print_exc()

    return jsonify({

        "status": "error",

```

```
        "message":  
            str(e)  
    }, 500
```

4.8 Implementasi Aktivitas Fisik

implementasi fitur aktivitas fisik pada aplikasi *Heartify* . Data aktivitas yang dilakukan pengguna, seperti jenis aktivitas, durasi, dan waktu aktivitas, dikirimkan ke *backend* melalui *API* untuk disimpan ke dalam database. Data tersebut digunakan untuk membantu pengguna memantau riwayat aktivitas fisik sebagai bagian dari pencegahan risiko penyakit jantung.

```
from database.mongo import db  
from bson.objectid import ObjectId  
from datetime import datetime, timedelta  
  
class AktivitasModel:  
    collection = db["aktivitas_fisik"]  
  
    @staticmethod  
    def save_activity(user_id, langkah, jarak, durasi, kalori):  
        AktivitasModel.collection.insert_one({  
            "user_id": ObjectId(user_id),  
            "langkah": langkah,  
            "jarak": jarak,  
            "durasi": durasi,  
            "kalori": kalori,  
            "tanggal": datetime.utcnow()  
        })
```

4.9 Implementasi Fitur Pengingat

implementasi fitur pengingat pada aplikasi *Heartify* . Sistem menerima data berupa judul pengingat, waktu, dan jenis pengingat dari pengguna, kemudian

mengirimkan data tersebut ke *backend* melalui *API*. Data pengingat disimpan pada database agar dapat ditampilkan kembali dan membantu pengguna menjaga konsistensi pola hidup sehat.

```
@app.route('/add-reminder', methods=['POST'])
@token_required
def add_reminder():
    try:
        user_id = request.user["user_id"]

        data = request.get_json()

        reminders = db["reminders"]

        reminders.insert_one({
            "user_id": user_id,
            "title": data.get("title"),
            "category": data.get("category"),
            "hour": data.get("hour"),
            "minute": data.get("minute"),
            "is_active": data.get("is_active", True),
            "created_at": datetime.utcnow()
        })
    }
```