LAMPIRAN

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama	: Maulana Alamsyah
NIM	: 20090137
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	: -

Pada hari ini Senin tanggal 25 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan **syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua.** Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 25 Maret 2024

Pihak Pertama	Pihak Kedua
Maulana Alamsyah	Hepatika Zidny Ilmadina, S.Pd., 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 di bawah ini :

Pihak Pertama

Nama	: Maulana Alamsyah
NIM	: 20090137
Program Studi	: Teknik Informatika

Pihak Kedua

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

Pada hari ini Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat adanya kemajuan pengerjaan skripsi yang dipresentasikan setiap dua minggu. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama	Pihak Kedua
Maulana Alamsyah	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 PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Maulana Alamsyah
   Kewarganegaraan : Indonesia
   Alamat : JL. KH. Umar Asnawi Desa Kebasen RT 08 RW 02 Kecamatan Talang, Kabupaten Tegal, 52193
2. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Jalan kenanga Gang 1 Nomor 9, Kelurahan Mangkukusuman, Kecamatan Tegal Timur Kota Tegal
3. Nama : Dwi Intan Af'idah, S.T., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Desa Grinting RT003/RW001, Kecamatan Bulakamba, Kabupaten Brebes, 52253

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
   Berupa : Program Komputer
   Berjudul : Pengembangan Aplikasi Preg-Fit Berbasis Mobile Untuk Membantu Ibu Hamil Melakukan Senam Yoga Prenatal

   - 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, 22 Juli 2024

(Maulana Alamsyah)
Pemegang Hak Cipta *

(Hepatika Zidny Ilmadina, S.Pd., M.Kom.)
Pemegang Hak Cipta *

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

* Semua pemegang hak cipta agar menandatangani di atas materai.

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : Maulana Alamsyah
   Kewarganegaraan : Indonesia
   Alamat : JL. KH. Umar Asnawi Desa Kebasen RT 08 RW 02 Kecamatan Talang, Kabupaten Tegal, 52193.
2. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Jalan kenanga Gang 1 Nomor 9, Kelurahan Mangkukusuman, Kecamatan Tegal Timur Kota Tegal
3. Nama : Dwi Intan Af'idah, S.T., M.Kom.
   Kewarganegaraan : Indonesia
   Alamat : Desa Grinting RT003/RW001, Kecamatan Bulakamba, Kabupaten Brebes, 52253

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

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

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Pengembangan Aplikasi Preg-Fit Berbasis Mobile Untuk Membantu Ibu Hamil Melakukan Senam Yoga Prenatal". 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, 22 Juli 2024

Pemegang Hak Cipta

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

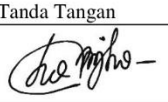
Pencipta

METERAI TEMPEL
C8FD6ALX311810369

(Maulana Alamsyah)

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

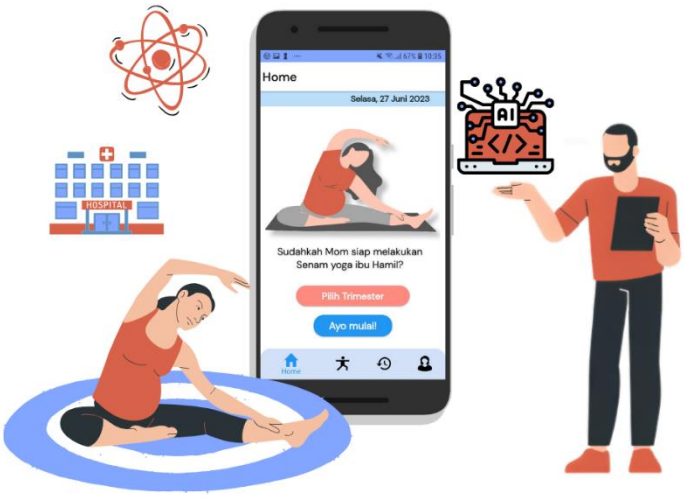
(Dwi Intan Af'idah, S.T., M.Kom.)

PESERTA PENGUJIAN PENERIMAAN PENGGUNA			
No	Nama	No. Kontak	Tanda Tangan
1.	Lusi Wahyuni	081327959713	
2.	Meyliya Qudriani, S.ST, M.Kes	085786222334	

MANUAL BOOK

Manual Book

Aplikasi Preg-Fit



Oleh:
Maulana Alamsyah
Hepatika Zidny Ilmadina, S.Pd., M.Kom
Dwi Intan Af'idah, S.T., M.Kom..

Pendahuluan

1. Latar Belakang

Kehamilan menyebabkan perubahan fisik dan psikologis pada ibu hamil. Senam yoga untuk ibu hamil adalah cara efektif untuk menjaga kesehatan dengan memperkuat otot-otot penting dan meningkatkan elastisitas tubuh. Manfaat lain dari senam yoga termasuk memperlancar sirkulasi darah, memperbaiki postur tubuh, mengurangi stres, dan melatih pernapasan.

Namun, data menunjukkan bahwa banyak ibu hamil kurang siap secara fisik dan psikologis untuk menghadapi persalinan. Ibu hamil yang tidak rutin melakukan senam yoga memiliki risiko lebih tinggi mengalami nyeri punggung. Selain itu, gerakan senam yoga harus disesuaikan dengan usia kehamilan atau trimester.

Untuk mengatasi kendala finansial dan memastikan pengawasan yang baik, penelitian ini mengembangkan aplikasi bernama Preg-Fit. Aplikasi ini dirancang untuk mendeteksi gerakan senam yoga bagi ibu hamil berdasarkan trimester kehamilan. Ibu hamil dapat menggunakan aplikasi ini secara mandiri, kapan saja dan di mana saja, tanpa perlu instruktur. Tujuannya adalah membantu ibu hamil yang kurang siap secara fisik dan psikologis menghadapi persalinan.

Dengan demikian, aplikasi Preg-Fit diharapkan dapat memberikan solusi yang efektif dan praktis bagi ibu hamil dalam menjaga kesehatan selama kehamilan dan mempersiapkan diri secara fisik dan psikologis untuk proses persalinan. Aplikasi ini terutama bermanfaat bagi mereka yang tidak memiliki akses ke kelas yoga antenatal atau instruktur senam, serta memberikan solusi yang mudah diakses untuk menjaga kesehatan dan kebugaran selama masa kehamilan.

2. Tujuan Pembuatan Dokumen

Dokumen manual book Aplikasi Preg-Fit ini dibuat dengan tujuan untuk mempermudah pengguna dalam menggunakan Aplikasi Preg-Fit. Dokumen manual book aplikasi ini berisi informasi yang menggambarkan dan penggunaan aplikasi.

3. Nilai Inovasi

Aplikasi senam yoga ibu hamil ini memiliki nilai inovasi yang tinggi dengan fitur pendeteksi gerakan senam yoga yang membantu ibu hamil dalam melakukan gerakan yang tepat dan menghindari gerakan berisiko. Selain itu, aplikasi ini juga memberikan aksesibilitas dan kemandirian kepada ibu hamil, memungkinkan mereka untuk melakukan senam yoga kapan saja dan di mana saja tanpa terkendala oleh biaya atau waktu. Dengan demikian, aplikasi ini memiliki potensi besar untuk meningkatkan kesehatan dan kesejahteraan ibu hamil, serta membantu persiapan fisik dan psikologis yang optimal dalam menghadapi persalinan.

4. Dampak Pemanfaatan Aplikasi Preg-Fit

Terdapat beberapa dampak pemanfaatan Aplikasi Preg-Fit, di antaranya :

- Menjaga kesehatan pada ibu hamil dan bayi selama masa kehamilan.
- Meningkatkan kesiapan fisik dan psikologis ibu hamil dalam menghadapi persalinan.
- Memberikan kemandirian kepada ibu hamil untuk melakukan senam yoga kapan saja dan di mana saja.
- Meminimalisir risiko gerakan yang salah dan berbahaya bagi ibu hamil melalui fitur pendeteksi gerakan.
- Memberikan fleksibelitas pada ibu hamil saat melakukan senam yoga hamil tanpa keterbatasan uang dan waktu.

- Meningkatkan kesehatan dan kesejahteraan ibu hamil dengan melancarkan sirkulasi darah, memperbaiki postur tubuh, dan mengurangi keluhan fisik.
- Peningkatan pengetahuan dan kesadaran ibu hamil tentang senam yoga hamil dan manfaatnya.
- Mengurangi stres dan kecemasan yang terkait dengan kehamilan melalui praktik senam yoga.
- Meningkatkan kemampuan pernapasan ibu hamil yang berguna saat proses persalinan.
- Menyediakan solusi self-help yang mendukung proses kehamilan hingga persalinan.

Pembahasan Aplikasi

1. **Perangkat yang dibutuhkan**
   - **Perangkat Lunak**
     Perangkat lunak yang digunakan adalah android operating system
   - **Perangkat Keras**
     Perangkat keras yang digunakan hanya *smartphone*
2. **Deskripsi Fungsional Aplikasi Preg-Fit**
   - **Tujuan utama**
     Aplikasi dapat menjadi solusi untuk membantu ibu hamil menjaga kesehatan ibu dan bayi saat mempersiapkan fisik dan prikologis dalam menghadapi persalinan, terutama bagi mereka yang tidak memiliki akses ke kelas yoga antenatal atau instruktur senam. Selain itu aplikasi ini memberikan solusi praktis dan mudah diakses bagi ibu hamil agar tetap bisa menjaga kesehatan dan kebugaran mereka selama masa kehamilan.
   - **Menu**
     1. Home
     2. Yoga
     3. History
     4. Profil
3. **Penjelasan Detail Fitur**

   Semua fitur pada Aplikasi Preg-Fit hanya dapat digunakan apabila terhubung dengan internet, Terdapat beberapa fitur pada aplikasi Preg-Fit diantaranya :
   - Fitur Daftar
     Pada fitur daftar pengguna dapat membuat akun, Ibu hamil dapat membuat akun dengan mengklik "Daftar yuk!" dan memasukkan nomor HP. Kemudian, aplikasi akan menampilkan pop up terkait apakah ibu hamil sudah mendapatkan izin dari dokter. Jika sudah, aplikasi akan meminta ibu hamil menyetujui pernyataan terkait kondisi kehamilannya. Jika belum, aplikasi akan mengarahkan ke halaman chatbot, yang akan mengajukan pertanyaan terkait kondisi kesehatan ibu hamil. Hasilnya, chatbot akan memberi tahu apakah ibu hamil diperbolehkan mendaftar atau tidak.

- Fitur Login

  Pada fitur login pengguna dapat masuk ke aplikasi, terdapat halaman yang menampilkan informasi untuk memasukkan nomor HP ibu hamil dan verifikasi OTP.

- Fitur Ubah Nomor HP

  Fitur ini terdapat pada halaman login, fitur ini memungkinkan ibu hamil untuk mengganti nomor HP baru dan diverifikasi menggunakan kode OTP yang dikirim ke email pengguna yang sudah terdaftar.

- Fitur Ubah Profil

  Fitur ini terdapat pada halaman profil, fitur ini memungkinkan ibu hamil untuk mengganti data, seperti nama, tanggal lahir, usia kandungan, dan email.

- Fitur Deteksi Senam Yoga Hamil Menyesuaikan Usia Kandungan

  Fitur ini dilakukan secara real-time sehingga ibu hamil dapat melakukan senam yoga hamil secara mandiri, aman, dan fleksibel waktu. Fitur ini terdapat pada halaman home, terdapat tombol "Pilih Trimester", ibu hamil dapat memilih usia kandungan, untuk memulai senam yoga ibu hamil dapat klik tombol "Ayo mulai!", selanjutnya ibu hamil akan diarahkan ke halaman kamera, terdapat intruksi gerakan senam yoga yang akan dilakukan, ibu hamil dapat mengikuti instruksi tersebut, gerakan ibu hamil akan di deteksi apakah gerakan benar atau salah, apabila salah melakukan gerakan Aplikasi Preg-Fit akan memberikan peringatan berupa suara.

- Fitur Deteksi Senam Yoga Hamil Pilih Jenis Gerakan Yoga

  Fitur ini dilakukan secara real-time sehingga ibu hamil dapat melakukan senam yoga hamil secara mandiri, aman, dan fleksibel waktu. Fitur ini terdapat pada halaman yoga, terdapat informasi pada card yaitu gambar dan jenis gerakan, ibu hamil dapat memilih jenis gerakan yang ingin dilakukan, selanjutnya ibu hamil akan diarahkan ke halaman kamera, terdapat intruksi gerakan senam yoga yang akan dilakukan, ibu hamil dapat mengikuti instruksi tersebut, gerakan ibu hamil akan di deteksi apakah gerakan benar atau salah, apabila salah melakukan gerakan Aplikasi Preg-Fit akan memberikan peringatan berupa suara.

- Fitur Feedback

  Fitur ini akan ditampilkan setelah ibu hamil menyelesaikan latihan senam yoga, fitur ini dapat diisi oleh ibu hamil sebagai evaluasi bagi developer.

- Fitur History

  Visualisasi data ditampilkan secara dinamis. Tujuannya adalah menampilkan tiga gerakan terpopuler yang sering dilakukan oleh seluruh pengguna Aplikasi Preg-Fit. Selain itu fitur history ini dapat menampilkan informasi aktifitas senam yoga hamil.

- Fitur Mengatur Pengingat

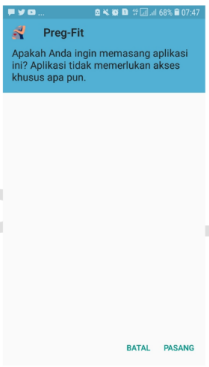
  Fitur pengingat dalam Aplikasi Preg-Fit dirancang untuk membantu ibu hamil mengatur jadwal latihan yoga secara teratur. Dengan fitur ini, ibu hamil dapat menetapkan pengingat atau notifikasi sesuai dengan waktu yang mereka pilih untuk melakukan senam yoga hamil. Fitur ini tidak hanya membantu dalam menjaga konsistensi latihan, tetapi juga memastikan bahwa ibu hamil tidak melewatkan sesi yang penting untuk kesehatan mereka. Ibu hamil dapat menambahkan, mengubah, atau menghapus pengingat sesuai kebutuhan mereka. Pengaturan pengingat ini diharapkan dapat membantu ibu hamil dalam menjaga kesehatan dan mempersiapkan diri untuk persalinan dengan lebih baik.

4. Penggunaan Aplikasi Preg-Fit

a. Cara Install Aplikasi Preg-Fit

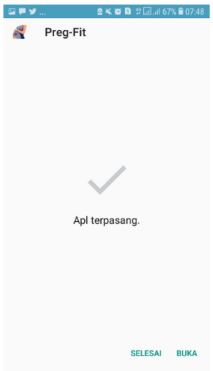
Aplikasi Preg-Fit belum terdaftar pada Play Store sehingga masih menggunakan file .apk, berikut merupakan langkah-langkah untuk menginstall Aplikasi Preg-Fit :

- Pastikan file .apk sudah tersedia pada *smartphone,* klik aplikasi tersebut dan layar akan menampilkan dialog konfirmasi pemasangan aplikasi.



Gambar 1 Dialog Pemasangan Aplikasi

- Selanjutnya ibu hamil dapat mengklik tombol pasang, yang selanjutnya masuk pada halaman pemasangan aplikasi.



Gambar 2 ProsesPemasangan Aplikasi

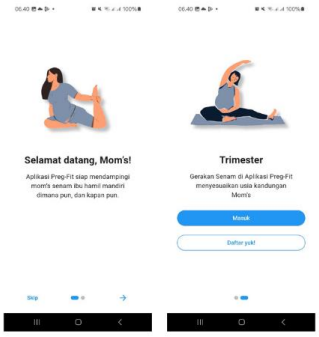
b. Cara Pengguna Daftar Dalam Aplikasi Preg-Fit

Berikut ini merupakan cara pengguna daftar dalam aplikasi Preg-Fit:

- Pastikan *smartphone* terhubung dengan internet, kemudian buka Aplikasi Preg-Fit:

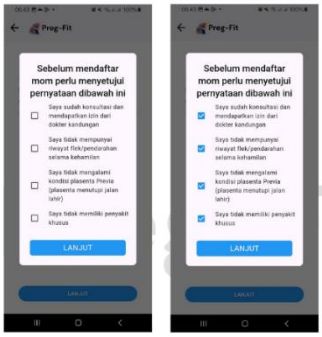
Gambar 3 Tampilan masuk aplikasi / Halaman SplashScreen

- Setelah membuka aplikasi, akan masuk ke halaman splash screen. Jika ini adalah pertama kalinya pengguna menginstal aplikasi Preg-Fit, maka akan muncul sebuah pop-up yang meminta izin untuk mengirim notifikasi. Ibu hamil dapat klik Izinkan, kemudian ibu hamil akan diarahkan ke halaman onboarding, halaman ini menampilkan informasi singkat mengenai Aplikasi Preg-Fit. Halaman ke dua dari onboarding, terdapat tombol untuk login dan daftar. Untuk mendaftar aplikasi ibu hamil dapat memilih tombol "Daftar yuk!"



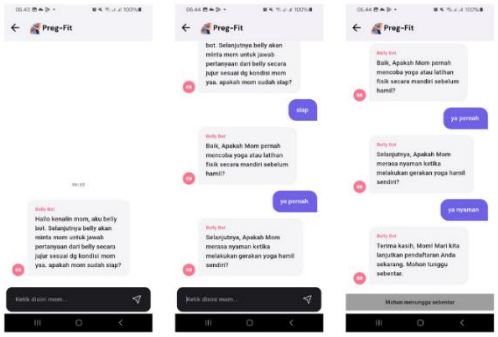
Gambar 4 Halaman Onboarding

- Ibu hamil dapat membuat akun dengan mengklik "Daftar yuk!" dan memasukkan nomor HP. Kemudian, aplikasi akan menampilkan pop-up terkait apakah ibu hamil sudah mendapatkan izin dari dokter.
- Jika sudah, aplikasi akan meminta ibu hamil menyetujui pernyataan terkait kondisi kehamilannya. Jika belum, aplikasi akan mengarahkan ke halaman chatbot. Hasilnya, chatbot akan memberi tahu apakah ibu hamil diperbolehkan mendaftar atau tidak.Berikut merupakan jika ibu hamil sudah mendapatkan izin dari dokter untuk melakukan senam yoga hamil.



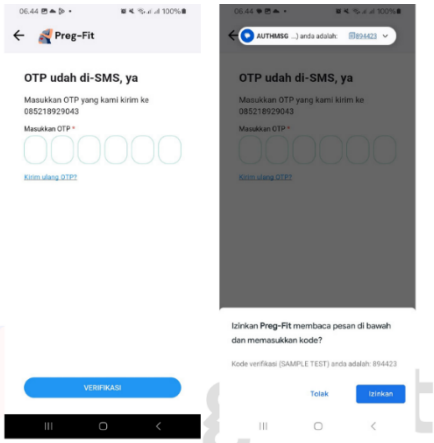
Gambar 5 Halaman Daftar Sudah Mendapat Izin Dari Dokter

- Berikut merupakan tampilan jika ibu hamil belum mendapatkan izin dari dokter untuk melakukan senam yoga hamil.



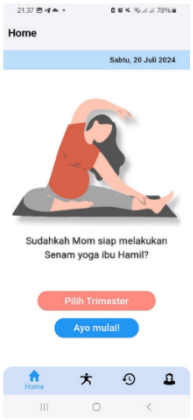
Gambar 6 Halaman Daftar Belum Mendapat Izin Dari Dokter

- Langkah selanjutnya adalah verifikasi OTP yang dikirimkan ke SMS, ibu hamil dapat memasukkan kode OTP yang didapat.



Gambar 7 Halaman Verifikasi

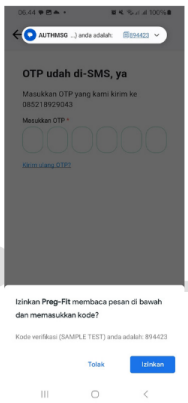
- Setelah nomor HP berhasil di verifikasi, ibu hamil akan diarahkan ke halaman home



Gambar 8 Halaman Home

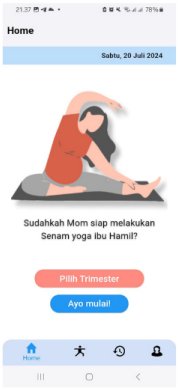
c. Cara Pengguna Masuk Dalam Aplikasi Preg-Fit

- Setelah berhasil mendaftar, ibu hamil dapat membuka aplikasi, di halaman ke dua onboarding, ibu hamil dapat memilih tombol "Masuk". Kemudian ibu hamil akan diarahkan pada halaman masukan nomor HP dan klik lanjut, aplikasi mengirimkan SMS berisi kode OTP



Gambar 9 Verifikasi Nomor HP

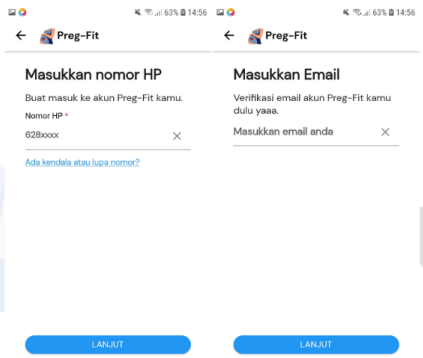
- Setelah nomor HP berhasil di verifikasi, ibu hamil akan diarahkan ke halaman home



Gambar 10 Halaman Home

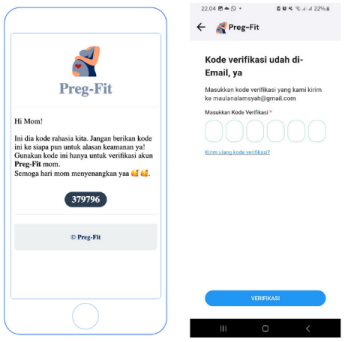
d. Cara Pengguna Lupa Nomor HP Aplikasi Preg-Fit

- Ibu hamil dapat memilih tombol "Masuk". Kemudian ibu hamil akan diarahkan pada halaman masukan nomor HP, apabila ibu hamil lupa nomor HP, maka ibu hamil dapat klik tautan "Ada kendala atau Lupa nomor?" yang selanjutnya akan di arahkan untuk memasukan email di halaman masukan email, selanjutnya ibu hamil dapat klik tombol "LANJUT"



Gambar 10 Halaman Masukan Nomor HP dan Masukan Email

- Selanjutnya akan terdapat pop up berisi informasi mengenai kode OTP di kirim melalui email. kemudian akan diarahkan ke halaman verifikasi OTP, ibu hamil dapat memasukkan kode OTP yang didapat dari email.

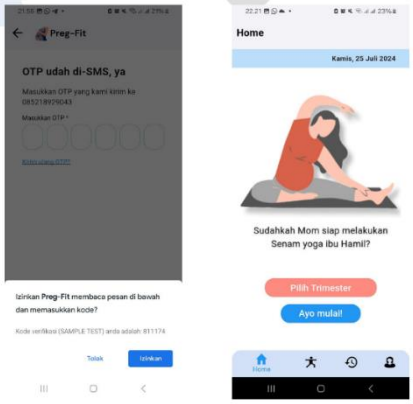


Gambar 11 Halaman Kode Verifikasi Email

- Selanjutnya akan masuk ke halaman ubah nomor HP

Gambar 12 Halaman Ubah Nomor HP

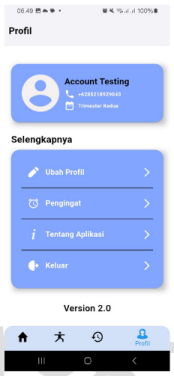
- Selanjutnya setelah mengisi nomor HP baru akan masuk pada halaman verifikasi OTP dan setelah diverifikasi akan masuk pada halaman home



Gambar 12 Halaman Ubah Nomor HP

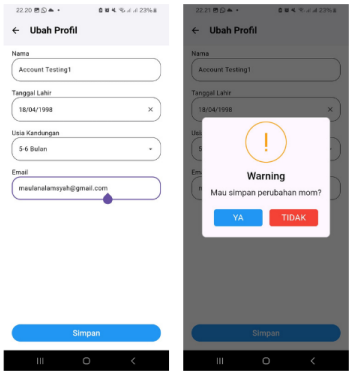
e. Cara Pengguna Ubah Profil

- Setelah berhasil mendaftar, ibu hamil dapat membuka aplikasi dan pastikan sudah login ke dalam Aplikasi Preg-Fit, untuk mengubah profil ibu hamil dapat klik menu profil selanjutnya pilih "Ubah Profil"



Gambar 13 Halaman Profil

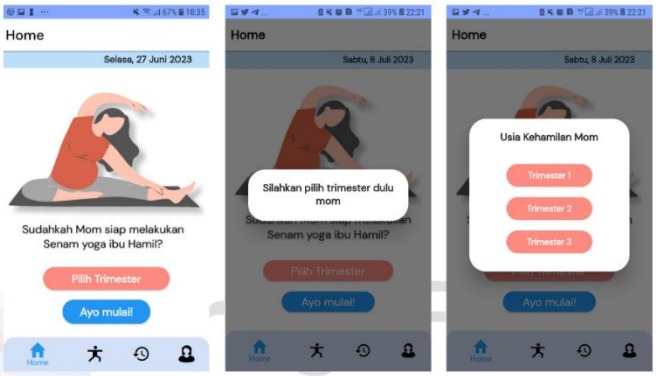
- Langkah selanjutnya ibu hamil diarahkan pada halaman ubah profil, setelah selesai mengubah profil, pengguna dapat klik simpan, saat pop up validasi muncul, ibu hamil dapat memilih apakah ingin melakukan perubahan atau tidak.



Gambar 14 Halaman Ubah Profil dan pop up Validasi

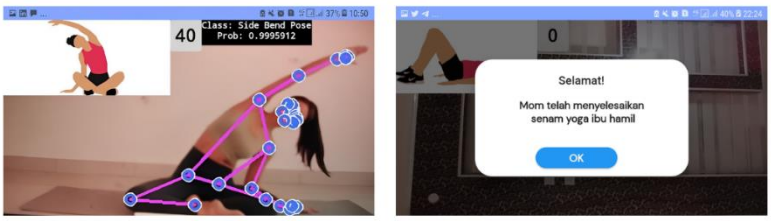
f. Cara Melakukan Senam Yoga Hamil Menyesuaikan Usia Kandungan

- Pada halaman home terdapat tombol "Pilih Trimester", ibu hamil dapat memilih menyesuaikan usia kandungan. Selanjutnya klik tombol "Ayo mulai!" untuk melakukan senam yoga hamil.



Gambar 15 Halaman Home

- Selanjutnya akan diarahkan pada halaman kamera untuk mendeteksi gerakan senam yoga, ibu hamil akan diberikan intruksi mengikuti gerakan pada layar HP sesuai dengan rangkaian gerakan trimester yang dipilih.



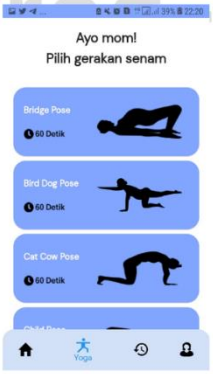
Gambar 16 Halaman Deteksi dan Pop up Informasi

- Setelah selesai melakukan rangkaian yoga, ibu hamil dapat memberikan feedback untuk aplikasi sebagai evaluasi developer.

Gambar 17 pop up Feedback

g. Cara Melakukan Senam Yoga Hamil Pilih Jenis Gerakan Yoga

- Pada halaman yoga, terdapat sebanyak delapan pilihan jenis yoga, ibu hamil dapat memilih jenis gerakan yang ingin dilakukan.

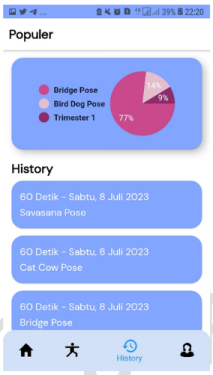


Gambar 18 Halaman Yoga

- Selanjutnya, ibu hamil akan diarahkan pada halaman kamera untuk melakukan senam yoga hamil sesuai dengan jenis gerakan yoga yang dipilih.
- Setelah selesai melakukan senam yoga hamil, aplikasi akan menampilkan pop up feedback.

h. Cara Melihat History Senam Yoga Hamil

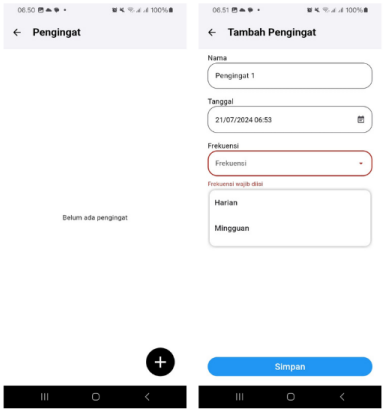
- Ibu hamil dapat memilih menu history, Visualisasi data yang dinamis memberikan informasi tiga gerakan senam yoga terpopuler. Selain itu fitur history dapat menampilkan informasi aktifitas senam yoga hamil.



Gambar 19 Halaman History

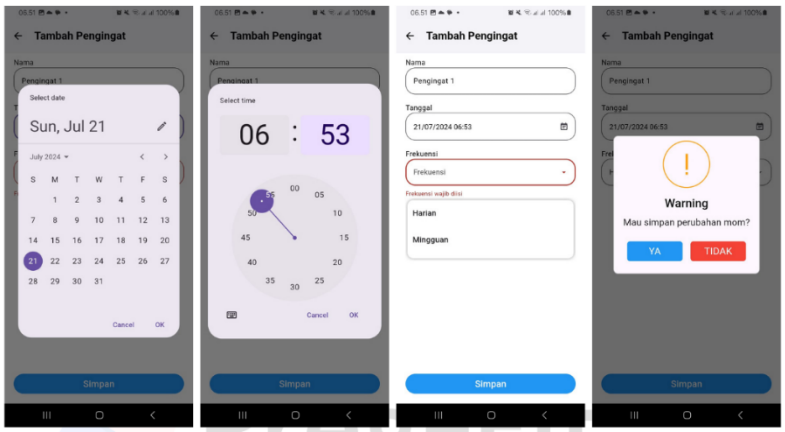
i. Cara Mengatur Pengingat

- Ibu hamil dapat memilih menu profil, kemudian klik buton "Pengingat"
- Diarahkan ke halaman pengingat, ibu hamil dapat menekan button tambah untuk menambahkan pengingat



Gambar 20 Halaman Pengingat dan Tambah Pengingat

- Mengisi judul, Tanggal dan Jam, serta Frekuensi

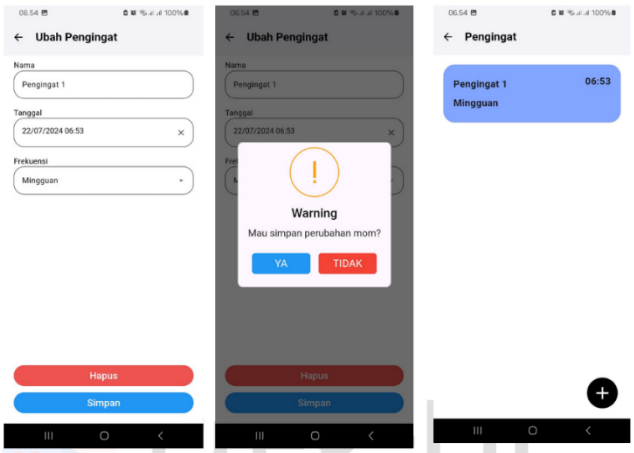


Gambar 21 Menambah Pengingat

- Pengingat yang ditambahkan

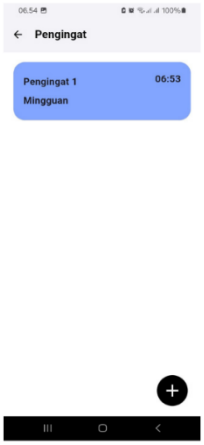
Gambar 22 Halaman Pengingat

- Mengubah Pengingat yang telah ditambahkan.



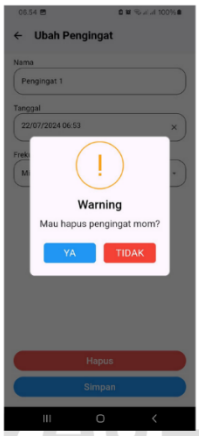
Gambar 23 Mengubah Pengingat

- Hasil pengingat yang diubah



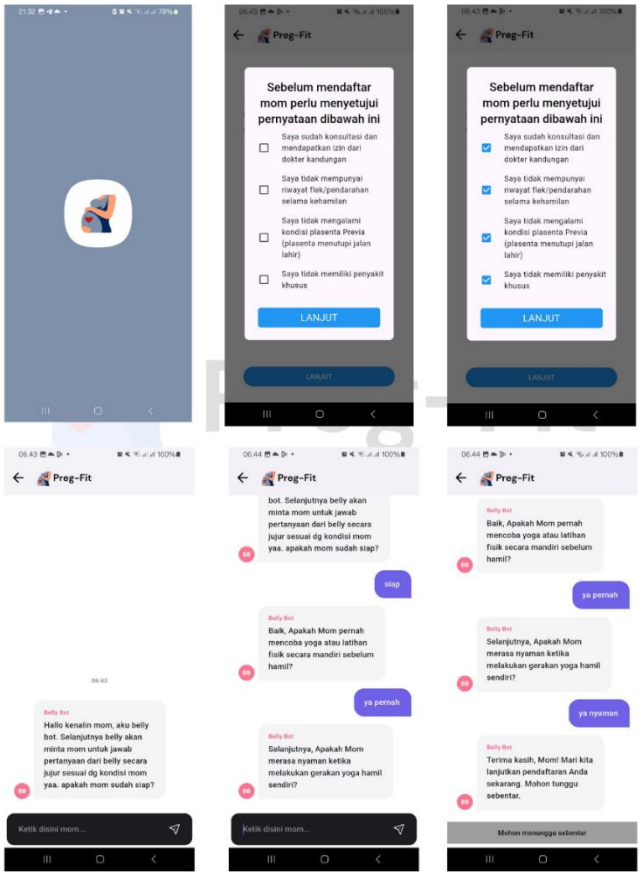
Gambar 24 Halaman Pengingat

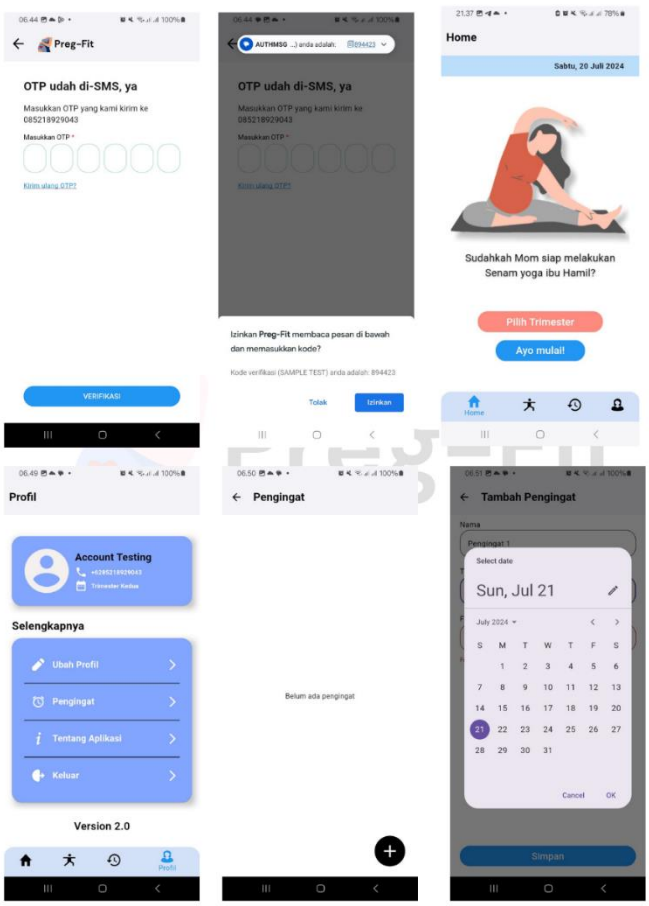
- Menghapus Pengingat yang telah ditambahkan.

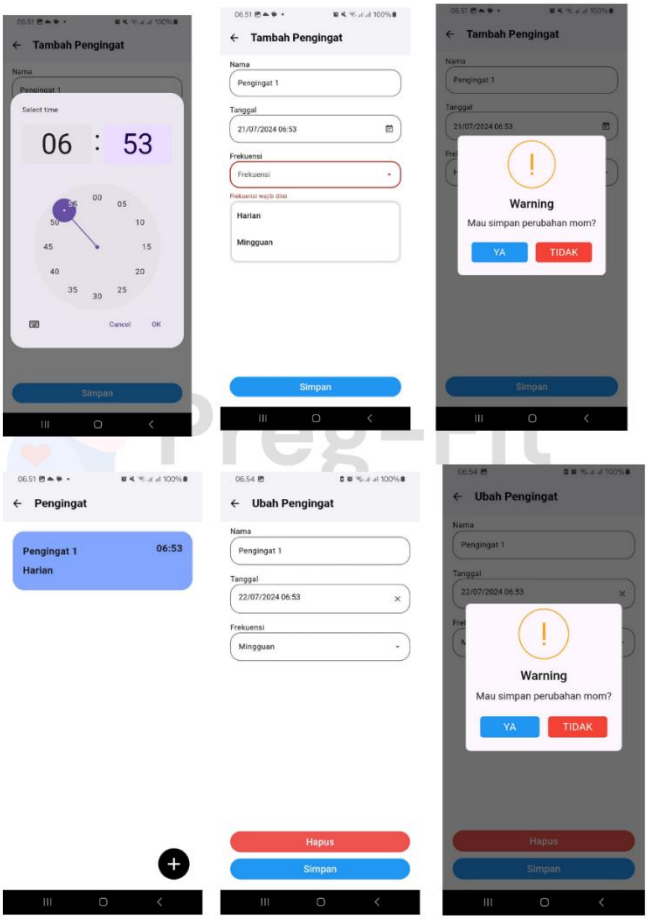


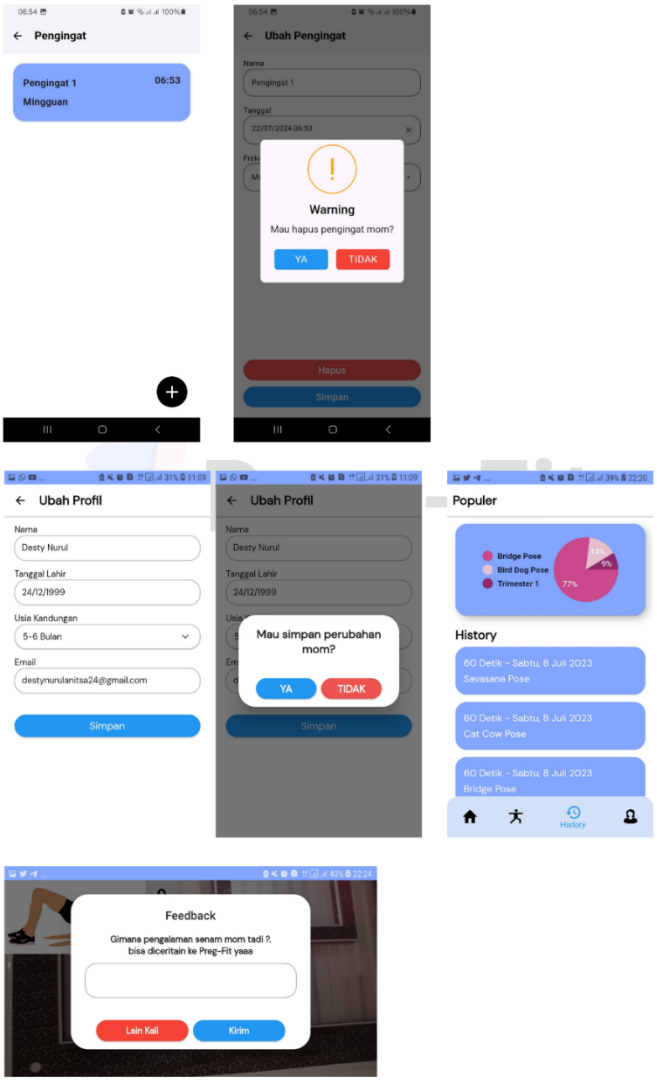
Gambar 25 Menghapus Pengingat

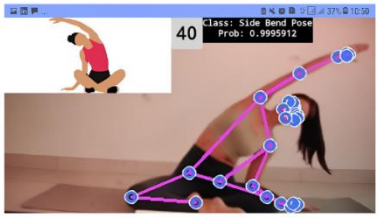
5. Screenshoot Aplikasi Preg-Fit



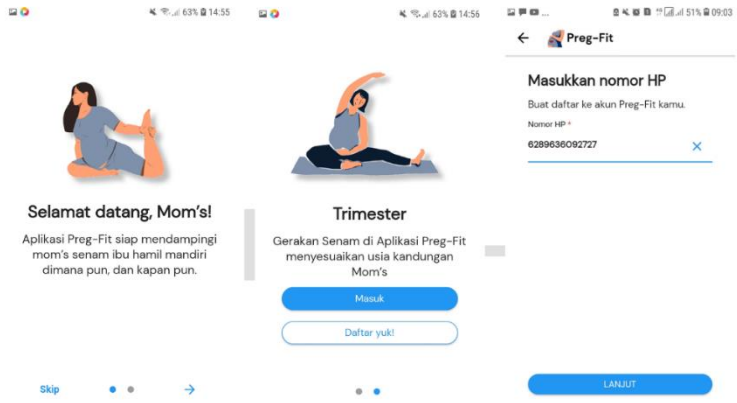

Preg-Fit
OTP udah di-SMS, ya
Masukkan OTP yang kami kirim ke
085218929043
Masukkan OTP *
Kirim ulang OTP?
VERIFIKASI
OTP udah di-SMS, ya
Izinkan Preg-Fit membaca pesan di bawah dan memasukkan kode?
Kode verifikasi (SAMPLE TEST) anda adalah: 894423
Tolak
Izinkan
Home
Sabtu, 20 Juli 2024
Sudahkah Mom siap melakukan Senam yoga ibu Hamil?
Pilih Trimester
Ayo mulai!
Profil
Account Testing
Selengkapnya
Ubah Profil
Pengingat
Tentang Aplikasi
Keluar
Version 2.0
Pengingat
Belum ada pengingat
Tambah Pengingat
Nama
Pengingat 1
Select date
Sun, Jul 21
July 2024
Cancel
OK
Simpan


Tambah Pengingat
Nama
Pengingat 1
Tanggal
21/07/2024 06:53
Frekuensi
Harian
Mingguan
Simpan
Warning
Mau simpan perubahan mom?
YA
TIDAK
Pengingat
Pengingat 1
06:53
Harian
Ubah Pengingat
Hapus


Pengingat
Pengingat 1
06:53
Mingguan
Ubah Pengingat
Nama
Pengingat 1
Tanggal
22/07/2024 06:53
Warning
Mau hapus pengingat mom?
YA
TIDAK
Hapus
Simpan
Ubah Profil
Nama
Desty Nurul
Tanggal Lahir
24/12/1999
Usia Kandungan
5-6 Bulan
Email
destynurulanitsa24@gmail.com
Simpan
Mau simpan perubahan mom?
YA
TIDAK
Populer
Bridge Pose
Bird Dog Pose
Trimester 1
History
60 Detik - Sabtu, 8 Juli 2023
Savasana Pose
60 Detik - Sabtu, 8 Juli 2023
Cat Cow Pose
60 Detik - Sabtu, 8 Juli 2023
Bridge Pose
Feedback
Gimana pengalaman senam mom tadi ?,
bisa diceritain ke Preg-Fit yaas
Lain Kali
Kirim


40
Class: Side Bend Pose
Prob: 0.9995912

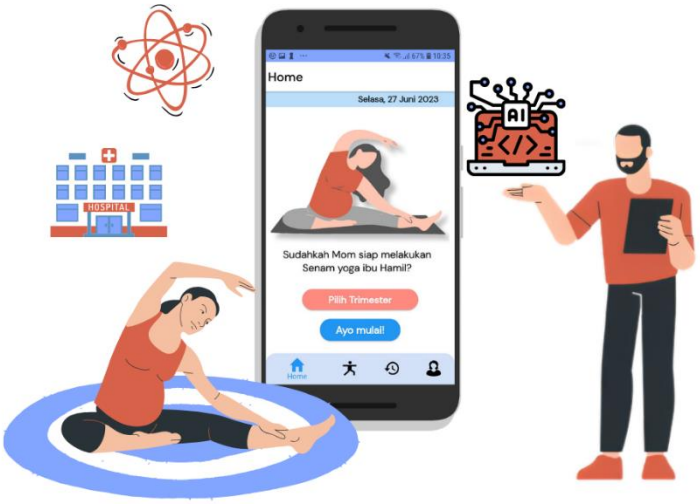

0
Selamat!
Mom telah menyelesaikan senam yoga ibu hamil
OK


Selamat datang, Mom's!
Aplikasi Preg-Fit siap mendampingi mom's senam ibu hamil mandiri dimana pun, dan kapan pun.
Skip
Trimester
Gerakan Senam di Aplikasi Preg-Fit menyesuaikan usia kandungan Mom's
Masuk
Daftar yuk!
Preg-Fit
Masukkan nomor HP
Buat daftar ke akun Preg-Fit kamu.
Nomor HP *
6289636092727
LANJUT

DOKUMEN TEKNIKAL

Dokumen Teknikal

Aplikasi Preg-Fit



Oleh:
Maulana Alamsyah
Hepatika Zidny Ilmadina, S.Pd., M.Kom
Dwi Intan Af'idah, S.T., M.Kom..

Pendahuluan

1. Latar Belakang

Kehamilan menyebabkan perubahan fisik dan psikologis pada ibu hamil. Senam yoga untuk ibu hamil adalah cara efektif untuk menjaga kesehatan dengan memperkuat otot-otot penting dan meningkatkan elastisitas tubuh. Manfaat lain dari senam yoga termasuk memperlancar sirkulasi darah, memperbaiki postur tubuh, mengurangi stres, dan melatih pernapasan.

Namun, data menunjukkan bahwa banyak ibu hamil kurang siap secara fisik dan psikologis untuk menghadapi persalinan. Ibu hamil yang tidak rutin melakukan senam yoga memiliki risiko lebih tinggi mengalami nyeri punggung. Selain itu, gerakan senam yoga harus disesuaikan dengan usia kehamilan atau trimester.

Untuk mengatasi kendala finansial dan memastikan pengawasan yang baik, penelitian ini mengembangkan aplikasi bernama Preg-Fit. Aplikasi ini dirancang untuk mendeteksi gerakan senam yoga bagi ibu hamil berdasarkan trimester kehamilan. Ibu hamil dapat menggunakan aplikasi ini secara mandiri, kapan saja dan di mana saja, tanpa perlu instruktur. Tujuannya adalah membantu ibu hamil yang kurang siap secara fisik dan psikologis menghadapi persalinan.

Dengan demikian, aplikasi Preg-Fit diharapkan dapat memberikan solusi yang efektif dan praktis bagi ibu hamil dalam menjaga kesehatan selama kehamilan dan mempersiapkan diri secara fisik dan psikologis untuk proses persalinan. Aplikasi ini terutama bermanfaat bagi mereka yang tidak memiliki akses ke kelas yoga antenatal atau instruktur senam, serta memberikan solusi yang mudah diakses untuk menjaga kesehatan dan kebugaran selama masa kehamilan.

2. Tujuan Pembuatan Dokumen

Dokumen Teknikal Aplikasi Preg-Fit ini dibuat dengan tujuan untuk mempermudah developer dalam maintance Aplikasi Preg-Fit. Dokumen teknis aplikasi ini berisi informasi yang menggambarkan dan penggunaan aplikasi.

3. Nilai Inovasi

Aplikasi senam yoga ibu hamil ini memiliki nilai inovasi yang tinggi dengan fitur pendeteksi gerakan senam yoga yang membantu ibu hamil dalam melakukan gerakan yang tepat dan menghindari gerakan berisiko. Selain itu, aplikasi ini juga memberikan aksesibilitas dan kemandirian kepada ibu hamil, memungkinkan mereka untuk melakukan senam yoga kapan saja dan di mana saja tanpa terkendala oleh biaya atau waktu. Dengan demikian, aplikasi ini memiliki potensi besar untuk meningkatkan kesehatan dan kesejahteraan ibu hamil, serta membantu persiapan fisik dan psikologis yang optimal dalam menghadapi persalinan.

4. Dampak Pemanfaatan Aplikasi Preg-Fit

Terdapat beberapa dampak pemanfaatan Aplikasi Preg-Fit, di antaranya :

- Menjaga kesehatan pada ibu hamil dan bayi selama masa kehamilan.
- Meningkatkan kesiapan fisik dan psikologis ibu hamil dalam menghadapi persalinan.
- Memberikan kemandirian kepada ibu hamil untuk melakukan senam yoga kapan saja dan di mana saja.
- Meminimalisir risiko gerakan yang salah dan berbahaya bagi ibu hamil melalui fitur pendeteksi gerakan.
- Memberikan fleksibelitas pada ibu hamil saat melakukan senam yoga hamil tanpa keterbatasan uang dan waktu.

5. Spesifikasi Teknis Aplikasi Preg-Fit

Spesifikasi Teknis meliputi :

- Manual Pengguna
- Source Code

Berikut uraian spesifikasi yang digunakan untuk membangun aplikasi :

- Python
- Web Browser
- Jupyter Notebook
- Visual Studio Code
- Xampp

Source Code

Pada source kode kali ini akan dijelaskan Sebagian besar kode yang digunakan untuk membuat aplikasi ini, terutama pada kode python yang digunakan pada aplikasi dan juga beberapa kode dari flutter. Berikut ini adalah penjelasan mengenai kode – kode tersebut :

1. Import Library

```
from flask import Flask, Blueprint, request, jsonify
from flask_restx import Resource, Api, Namespace, reqparse
from flask_sqlalchemy import SQLAlchemy
from flask_mail import Mail, Message
from werkzeug.security import generate_password_hash, check_password_hash
from sqlalchemy import func
from werkzeug.security import generate_password_hash, check_password_hash
from werkzeug.datastructures import FileStorage
from datetime  import date, datetime, timedelta
from json import dumps
from flask_cors import CORS
from flask_socketio import SocketIO, emit, join_room, leave_room
from chatbot import *
from dotenv import load_dotenv
from twilio.rest import Client
from twilio.base.exceptions import TwilioRestException
from transformers import BertTokenizer, TFBertForSequenceClassification
import base64
import cv2
import numpy as np
import mediapipe as mp
import tensorflow as tf
import time
import os
import imghdr
import re
import jwt
import random
import pytz
```

Gambar 1 Import Library

Kode di atas mengimpor berbagai library untuk membangun aplikasi web menggunakan Flask. Library yang digunakan mencakup Flask, Flask-Mail untuk pengiriman email, Flask-RESTx untuk membuat API RESTful, dan Flask-SQLAlchemy untuk integrasi basis data. Library SQLAlchemy digunakan untuk operasi basis data, sementara Werkzeug menyediakan utilitas keamanan untuk hash dan verifikasi kata sandi serta penanganan file yang diunggah. Pustaka standar Python seperti datetime dan json juga digunakan untuk operasi tanggal, waktu, dan serialisasi JSON. Flask-CORS diimpor untuk mengaktifkan Cross-Origin Resource Sharing (CORS). Library tambahan yang digunakan termasuk Flask-SocketIO untuk komunikasi real-time, MediaPipe untuk deteksi landmark, Keras dan TensorFlow untuk model pembelajaran mesin, JWT untuk token otentikasi, dan OpenCV untuk pemrosesan gambar. Pustaka lainnya seperti base64, numpy, random, dan pytz juga diimpor untuk berbagai utilitas tambahan.

2. Konfigurasi Varialbel Environment

```
SQL_USERNAME = os.getenv('SQL_USERNAME')
SQL_PASSWORD = os.getenv('SQL_PASSWORD')
SQL_DB = os.getenv('SQL_DB')
twilio_account_sid = os.getenv('TWILIO_SID')
twilio_auth_token = os.getenv('TWILIO_TOKEN')
twilio_services = os.getenv('TWILIO_SERVICES')
client = Client(twilio_account_sid, twilio_auth_token)
SECRET_KEY = os.getenv('APP_SECRET_KEY')
API_KEY = os.getenv('API_KEY')
ISSUER = "myFlaskWebService"
AUDIENCE_MOBILE = "myMobileApp"
blueprint = Blueprint('api', __name__, url_prefix='/api')
app.register_blueprint(blueprint)
app.config['MAIL_SERVER']= os.getenv('MAIL_SERVER')
app.config['MAIL_PORT'] = 2525
app.config['MAIL_USERNAME'] = os.getenv('MAIL_USERNAME')
app.config['MAIL_PASSWORD'] = os.getenv('MAIL_PASSWORD')
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USE_SSL'] = False
mail = Mail(app)
```

Gambar 2 Konfigurasi Variable Environment

Kode di atas menginisialisasi beberapa variabel dan layanan eksternal dalam aplikasi Flask dengan mengambil nilai-nilai dari variabel lingkungan menggunakan `os.getenv()`. Variabel terkait koneksi database (`SQL_USERNAME`, `SQL_PASSWORD`, `SQL_DB`), layanan Twilio (`twilio_account_sid`, `twilio_auth_token`, `twilio_services`), serta konfigurasi aplikasi (`SECRET_KEY`, `API_KEY`) diatur. Flask blueprint untuk API didaftarkan dengan URL prefix `/api`. Selain itu, konfigurasi server email diatur menggunakan variabel lingkungan untuk mengirim email melalui `Mail` dengan menggunakan protokol TLS. Twilio Client juga diinisialisasi untuk mengirim pesan.

3. Konfigurasi Request Parser

```
parser4OTPsend = reqparse.RequestParser()
parser4OTPsend.add_argument('no_hp', type=str, location='json', required=True, help='Nomor HP')

parser4OTPverif = reqparse.RequestParser()
parser4OTPverif.add_argument('no_hp', type=str, location='json', required=True, help='Nomor HP')
parser4OTPverif.add_argument('otp', type=str, location='json', required=True, help='OTP')
parser4OTPverif.add_argument('action', type=int, location='json', required=True, help='Aksi')
parser4OTPverif.add_argument('email', type=str, location='json', required=False, help='Email')

parser4ChatBot = reqparse.RequestParser()
parser4ChatBot.add_argument('message', type=str, location='json', required=True, help='Message')

parser4CheckEmail = reqparse.RequestParser()
parser4CheckEmail.add_argument('email', type=str, location='json', required=True, help='Email')

parser4UpdateUser = reqparse.RequestParser()
parser4UpdateUser.add_argument('no_hp', type=str, location='json', required=False, help='Nomor HP')
parser4UpdateUser.add_argument('email', type=str, location='json', required=False, help='Email')
parser4UpdateUser.add_argument('nama', type=str, location='json', required=False, help='Nama Lengkap')
parser4UpdateUser.add_argument('usia_kandungan', type=str, location='json', required=False, help='Usia Kandungan')
parser4UpdateUser.add_argument('tanggal_lahir', type=str, location='json', required=False, help='Tanggal Lahir')

parser4History = reqparse.RequestParser()
parser4History.add_argument('tanggal', type=str, location='json', required=True, help='Tanggal')
parser4History.add_argument('waktu', type=str, location='json', required=True, help='Waktu')
parser4History.add_argument('jenis_yoga', type=str, location='json', required=True, help='Jenis Yoga')

parser4Feedback = reqparse.RequestParser()
parser4Feedback.add_argument('komentar', type=str, location='json', required=True, help='Komentar')

parser4CheckNo = reqparse.RequestParser()
parser4CheckNo.add_argument('no_hp', type=str, location='json', required=True, help='Nomor HP')

parser4SendOtpMail = reqparse.RequestParser()
parser4SendOtpMail.add_argument('api-key', type=str, location='headers', required=True, help='API Key')
parser4SendOtpMail.add_argument('email', type=str, location='json', required=True, help='Email')

parser4VerifyOtpMail = reqparse.RequestParser()
parser4VerifyOtpMail.add_argument('api-key', type=str, location='headers', required=True, help='Api Key')
parser4VerifyOtpMail.add_argument('otp', type=str, location='json', required=True, help='OTP')
parser4VerifyOtpMail.add_argument('email', type=str, location='json', required=True, help='Email')
```

Gambar 3 Konfigurasi Request Parser

Kode tersebut mengonfigurasi beberapa `RequestParser` menggunakan `reqparse` dari Flask-RESTful untuk memvalidasi argumen yang diterima dalam berbagai endpoint API. Setiap parser dirancang untuk memeriksa tipe dan lokasi argumen yang diharapkan, seperti nomor HP untuk pengiriman OTP (`parser4OTPsend`), verifikasi OTP (`parser4OTPverif`), pesan chatbot (`parser4ChatBot`), dan pembaruan informasi pengguna (`parser4UpdateUser`). Parser lainnya termasuk `parser4History` untuk mencatat riwayat yoga, `parser4Feedback` untuk komentar, dan `parser4CheckNo` untuk pengecekan nomor HP. Selain itu, `parser4SendOtpMail` dan `parser4VerifyOtpMail` memvalidasi API key dan email untuk mengirim dan memverifikasi OTP melalui email. Penggunaan parser ini memastikan bahwa data yang diterima sesuai dengan format yang diharapkan dan memudahkan penanganan kesalahan.

4. Route dan API User

```python
@api.route('/users',methods=['GET', 'PUT'])
class User_Route(Resource):
    @api.response(200, 'OK')
    @api.doc(security='Bearer')
    def get(self):
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        try:
            payload = decode(jwtToken)
            user = db.session.execute(db.select(User).filter_by(id=payload['user_id'])).first()
            user = user[0]
            user.tanggal_lahir = dumps(user.tanggal_lahir, default=str).replace('"','')
        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401
        return {
            'id': user.id,
            'no_hp': user.no_hp,
            'email': user.email,
            'nama': user.nama,
            'usia_kandungan': user.usia_kandungan,
            'tanggal_lahir': user.tanggal_lahir
        }, 200
    @api.doc(security='Bearer')
    @api.expect(parser4UpdateUser)
    def put(self):
        print(request.json)
        try:
            args = request.get_json()
        except:
            return 1
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        no_hp = args['no_hp']
        email = args['email']
        nama = args['nama']
        if args['usia_kandungan']:
            usia_kandungan = args['usia_kandungan']
        if args['tanggal_lahir']:
            tanggal_lahir = args['tanggal_lahir']
            tanggal_lahir = tanggal_lahir.split("/")
            tanggal_lahir = date(int(tanggal_lahir[2]),int(tanggal_lahir[1]),int(tanggal_lahir[0]))
        try:
            payload = decode(jwtToken)
            userdb = db.session.execute(db.select(User).filter_by(id=payload['user_id'])).first()
            userdb = userdb[0]
        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401
        isUpdate = False
        if no_hp:
            userdb.no_hp = no_hp
            isUpdate = True
        if email:
            userdb.email = email
            isUpdate = True
        if nama:
            userdb.nama = nama
            isUpdate = True
        if args['usia_kandungan']:
            userdb.usia_kandungan = usia_kandungan
            isUpdate = True
        if args['tanggal_lahir']:
            userdb.tanggal_lahir = tanggal_lahir
            isUpdate = True
        if isUpdate:
            db.session.commit()
        return args, 200
```

Gambar 4 Route dan API User

Kode di atas mendefinisikan sebuah kelas `User_Route` yang mengelola endpoint `/users` dengan metode HTTP `GET` dan `PUT` dalam aplikasi Flask menggunakan Flask-RESTful. Kelas ini menggunakan dekorator `@api.route` untuk menentukan rute dan mendukung dua metode HTTP. Metode `GET` digunakan untuk mengambil informasi pengguna, sementara metode `PUT` digunakan untuk memperbarui informasi pengguna. Dekorator tambahan seperti `@api.response` dan `@api.doc` digunakan untuk mendefinisikan respons dan mengharuskan autentikasi Bearer token untuk mengakses endpoint ini.

Dalam metode `GET`, token JWT diambil dari header `Authorization` dan didekode untuk mengambil `user_id` dari payload. Kemudian, data pengguna diambil dari database berdasarkan `user_id`. Jika token valid, data pengguna seperti `id`, `no_hp`, `email`, `nama`, `usia_kandungan`, dan `tanggal_lahir` dikembalikan dalam format JSON dengan status kode 200. Jika token tidak valid, respons dengan pesan kesalahan dan status kode 401 dikembalikan. Penggunaan JWT memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses data ini.

Metode `PUT` digunakan untuk memperbarui informasi pengguna. Data JSON yang diterima divalidasi menggunakan `parser4UpdateUser`. Token JWT diambil dan didekode untuk mendapatkan `user_id`, yang kemudian digunakan untuk mengambil data pengguna dari database. Informasi pengguna diperbarui berdasarkan data yang diterima, seperti `no_hp`, `email`, `nama`, `usia_kandungan`, dan `tanggal_lahir`. Setelah memperbarui informasi, perubahan disimpan ke database dan data yang diterima dikembalikan dalam format JSON dengan status kode 200. Jika token tidak valid, respons dengan pesan kesalahan dan status kode 401 dikembalikan.

5. Route dan API Cek Nomor HP

```python
@api.route('/check_no')
class Check_No_Route(Resource):
    @api.expect(parser4CheckNo)
    @api.response(200, 'OK')
    def post(self):
        args = parser4CheckNo.parse_args()
        no_hp = args['no_hp']

        if no_hp.startswith("0"):
            no_hp = no_hp[1:]

        result = db.session.execute(db.select(User).filter((User.no_hp == f'62{no_hp}') | (User.no_hp == no_hp)))
        user = result.fetchone()
        if user:
            return {
                'message': 'Nomor HP sudah digunakan, silahkan langsung masuk aja mom!'
            }, 409

        return {
            'message' : 'Nomor hp belum terdaftar'
        }, 200
```

Gambar 5 Route dan API Cek Nomor HP

API ini mengelola endpoint `/check_no` yang digunakan untuk memeriksa apakah nomor HP sudah terdaftar dalam database. Endpoint ini hanya mendukung metode HTTP `POST` dan menggunakan `parser4CheckNo` untuk memvalidasi input dari permintaan JSON.

Saat permintaan `POST` diterima, API mengambil nomor HP dari input, menghapus awalan "0" jika ada, dan memeriksa keberadaan nomor tersebut dalam database dengan format internasional (menambahkan awalan "62") serta format lokal. Jika nomor HP sudah ada dalam database, API mengembalikan pesan dengan status kode 409, menandakan nomor tersebut sudah terdaftar. Sebaliknya, jika nomor HP belum terdaftar, API mengembalikan pesan dengan status kode 200.

6. Route dan API Kirim OTP

```python
@api.route('/send_otp')
class Send_OTP_Route(Resource):
    @api.expect(parser4OTPsend)
    @api.response(200, 'OK')
    def post(self):
        args = parser4OTPsend.parse_args()
        no_hp = format_phone_number(args['no_hp'])

        verification = client.verify.v2.services(twilio_services).verifications.create(to=no_hp, channel='sms')

        if verification.status == "pending":
            return {
                'message': 'OTP berhasil dikirim mom!'
            }, 200

        return {
            'message' : 'Gagal kirim OTP mom!'
        }, 500
```

Gambar 6 Route dan API Kirim OTP

API ini mengelola endpoint `/send_otp` yang digunakan untuk mengirim OTP (One-Time Password) melalui SMS. Endpoint ini hanya mendukung metode HTTP `POST` dan menggunakan `parser4OTPsend` untuk memvalidasi input yang diterima dalam permintaan JSON.

Saat permintaan `POST` diterima, API mengambil nomor HP dari input, memformatnya menggunakan fungsi `format_phone_number`, dan kemudian menggunakan layanan Twilio untuk mengirim OTP melalui SMS. Jika status verifikasi dari Twilio adalah "pending", yang menunjukkan bahwa OTP berhasil dikirim, API mengembalikan pesan dengan status kode 200. Jika terjadi kegagalan dalam pengiriman OTP, API mengembalikan pesan kesalahan dengan status kode 500.

7. Route dan API Verifikasi OTP

```python
@api.route('/verif_otp')
class Verif_OTP_Route(Resource):
    @api.expect(parser4OTPverif)
    @api.response(200, 'OK')
    def post(self):
        args = parser4OTPverif.parse_args()
        no_hp = format_phone_number(args['no_hp'])
        otp = args['otp']
        action = args['action']
        email = args.get('email')
        if action == 0:
            user = db.session.execute(db.select(User).filter(User.no_hp == no_hp)).scalar()
            if not user:
                return {'message': 'No HP mom belum terdaftar di Preg-Fit, mom bisa daftar dulu'}, 400
            valid, error_message = verif_otp(no_hp, otp)
            if valid:
                token = generate_token(user)
                return {'token': token}, 200
            return {'message': error_message}, 400
        elif action == 1:
            user = db.session.execute(db.select(User).filter_by(no_hp=no_hp)).scalar()
            if user:
                return {'message': 'Nomor HP sudah digunakan, silahkan langsung masuk aja mom!'}, 409
            valid, error_message = verif_otp(no_hp, otp)
            if valid:
                new_user = User(no_hp=no_hp, tanggal_lahir='1999-01-01')
                db.session.add(new_user)
                db.session.commit()
                new_user = db.session.execute(db.select(User).filter(User.no_hp == no_hp)).scalar()
                token = generate_token(new_user)
                return {'token': token, 'message': 'Berhasil daftar mom'}, 201
            return {'message': error_message}, 400
        elif action == 2:
            user = db.session.execute(db.select(User).filter(User.no_hp == no_hp)).scalar()
            if user:
                return {'message': 'Nomor HP sudah digunakan, silahkan coba nomor lain!'}, 409
            valid, error_message = verif_otp(no_hp, otp)
            if valid:
                current_user = db.session.execute(db.select(User).filter_by(email=email)).scalar()
                current_user.no_hp = no_hp
                db.session.commit()
                token = generate_token(current_user)
                return {'token': token, 'message': 'No HP berhasil diubah'}, 200
            return {'message': error_message}, 400
```

Gambar 7 Route dan API Verifikasi OTP

API ini mengelola endpoint `/verif_otp` yang digunakan untuk memverifikasi OTP (One-Time Password) dan melakukan berbagai aksi tergantung pada nilai `action` yang diberikan. Jika `action` adalah 0, API memeriksa apakah nomor HP terdaftar dan, jika OTP valid, menghasilkan token autentikasi. Jika `action` adalah 1, API memeriksa keberadaan nomor HP, dan jika OTP valid, mendaftarkan nomor HP baru dan menghasilkan token. Jika `action` adalah 2, API memeriksa apakah nomor HP sudah digunakan, dan jika OTP valid, memperbarui nomor HP pengguna yang ada dan menghasilkan token. Respons yang dikembalikan tergantung pada hasil verifikasi OTP dan status nomor HP.

8. Route dan API Cek Email

```python
@api.route('/check_email')
class C_Email_Route(Resource):
    @api.expect(parser4CheckEmail)
    @api.response(200, 'OK')
    def post(self):
        args = parser4CheckEmail.parse_args()
        email = args['email']

        user = db.session.execute(db.select(User).filter_by(email=email)).first()
        if user:
            return {
                'message': 'Email terdaftar!'
            }, 409

        return {
            'message' : 'Email tidak terdaftar'
        }, 200
```

Gambar 8 Route dan API Cek Email

API ini mengelola endpoint `/check_email` yang digunakan untuk memeriksa apakah alamat email sudah terdaftar dalam database. Endpoint ini hanya mendukung metode HTTP `POST` dan menggunakan `parser4CheckEmail` untuk memvalidasi input yang diterima dalam permintaan JSON.

Saat permintaan `POST` diterima, API mengambil alamat email dari input dan melakukan query ke database untuk memeriksa keberadaan email tersebut. Jika email ditemukan dalam database, API mengembalikan pesan dengan status kode 409, menandakan bahwa email sudah terdaftar. Jika email tidak terdaftar, API mengembalikan pesan dengan status kode 200.

9. Chatbot.py

```
import json
import random
import pandas as pd
import regex as re
from sklearn.preprocessing import LabelEncoder
import nltk


# Package sentence tokenizer
nltk.download('punkt')
# Package lemmatization
nltk.download('wordnet')
# Package multilingual wordnet data
nltk.download('omw-1.4')

# Importing the dataset
with open('dataset/datasets.json') as content:
  data1 = json.load(content)

# Mendapatkan semua data ke dalam list
tags = [] # data tag
inputs = [] # data input atau pattern
responses = {} # data respon
words = [] # Data kata
classes = [] # Data Kelas atau Tag
documents = [] # Data Kalimat Dokumen
ignore_words = ['?', '!'] # Mengabaikan tanda spesial karakter

for intent in data1['intents']:
  responses[intent['tag']]=intent['responses']
  for lines in intent['patterns']:
    inputs.append(lines)
    tags.append(intent['tag'])
    for pattern in intent['patterns']:
      w = nltk.word_tokenize(pattern)
      words.extend(w)
      documents.append((w, intent['tag']))
      # add to our classes list
      if intent['tag'] not in classes:
        classes.append(intent['tag'])

# Konversi data json ke dalam dataframe
data = pd.DataFrame({"patterns":inputs, "tags":tags})

labelencoder = LabelEncoder()
data['tags'] = labelencoder.fit_transform(data['tags'])

def text_preprocessing(text):
  text = text.lower()                                  # Mengubah teks menjadi lower case
  text = re.sub(r'https?://\S+|www\.\S+', '', text) # Menghapus URL
  text = re.sub(r'[-+]?[0-9]+', '', text)           # Menghapus angka
  text = re.sub(r'[^\w\s]','', text)                # Menghapus karakter tanda baca
  text = text.strip()                                  # Menghapus whitespaces
  return text

data['patterns'] = data['patterns'].apply(text_preprocessing)
```

Gambar 8 Chatbot.py

Kode ini mempersiapkan dataset untuk model chatbot dengan mengimpor dan memproses data dari file JSON yang berisi pola pertanyaan dan respons. Dataset diolah untuk menghasilkan daftar kata, dokumen, dan kelas, serta mengubah tag menjadi format numerik menggunakan `LabelEncoder`. Proses pembersihan teks dilakukan untuk menghapus URL, angka, tanda baca, dan spasi ekstra, menggunakan fungsi `text_preprocessing`. Hasilnya disimpan dalam DataFrame pandas, yang memudahkan manipulasi dan analisis data untuk pelatihan model chatbot.

10. Route dan API Chatbot

```python
@api.route('/chatbot')
class Chatbot_Route(Resource):
    @api.expect(parser4ChatBot)
    @api.response(200, 'OK')
    def post(self):
        args = parser4ChatBot.parse_args()
        message = args['message']

        input_text_tokenized = bert_tokenizer.encode(message,
                                            truncation=True,
                                            padding='max_length',
                                             max_length = 20,
                                            return_tensors='tf')

        bert_predict = bert_load_model(input_text_tokenized)
        bert_output = tf.nn.softmax(bert_predict[0], axis=-1)        # Softmax function untuk mendapatkan hasil klasifikasi
        output = tf.argmax(bert_output, axis=1)

        response_tag = labelencoder.inverse_transform(output.numpy().flatten())[0]
        response = random.choice(responses[response_tag])

        if response:
            return response, 200

        return {
            'message' : 'Maaf BellyBot tidak dapat melakukan verifikasi saat ini, silahkan coba beberapa saat lagi. Terima kasih',
            'value': 99
        }, 200
```

Gambar 10 Route dan API Chatbot

API ini mengelola endpoint `/chatbot` yang digunakan untuk mengirim pesan ke chatbot dan mendapatkan respons berdasarkan analisis pesan tersebut. Endpoint ini hanya mendukung metode HTTP `POST` dan menggunakan `parser4ChatBot` untuk memvalidasi input yang diterima dalam permintaan JSON.

Saat permintaan `POST` diterima, API mengambil pesan dari input dan memprosesnya menggunakan model BERT untuk klasifikasi. Pesan tersebut dikodekan dengan tokenizer BERT, kemudian model BERT yang telah dimuat digunakan untuk menghasilkan prediksi. Output dari model diolah dengan fungsi softmax untuk mendapatkan probabilitas klasifikasi, dan label yang paling mungkin dipilih menggunakan `tf.argmax`. Berdasarkan label hasil klasifikasi, chatbot memilih salah satu respons dari daftar respons yang telah ditetapkan.

Jika chatbot berhasil menghasilkan respons, API mengembalikan respons tersebut dengan status kode 200. Jika terjadi kegagalan dalam menghasilkan respons, API mengembalikan pesan kesalahan dengan status kode 200, menandakan bahwa chatbot tidak dapat melakukan verifikasi saat ini.

11. Route dan API Gerakan Populer

```
@api.route('/popular')
class Popular_Route(Resource):
    @api.doc(security='Bearer')
    def get(self):
        auth = request.headers.get('Authorization')

        jwtToken = auth[7:]

        try:
            payload = decode(jwtToken)
            payload = decode(jwtToken)

            user = db.session.execute(db.select(User).filter_by(id=payload['user_id'])).first()
            user = user[0]

            if user:
                count_query = db.session.query(History.jenis_yoga,
func.count(History.jenis_yoga).label('count')).group_by(History.jenis_yoga).order_by(func.count(History.jenis_yoga).desc())

                top_3_popular_jenis_yoga = count_query.limit(3).all()

                total_count = db.session.query(func.count(History.jenis_yoga)).scalar()

                result = []
                for jenis_yoga, count in top_3_popular_jenis_yoga:
                    popularity_percentage = int((count / total_count) * 100)
                    result.append({"jenis_yoga": jenis_yoga, "popularity_percentage": popularity_percentage})

        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401

        return result, 200
```

Gambar 11 Route dan API Gerakan Populer

API ini mengelola endpoint `/popular` yang digunakan untuk mendapatkan tiga jenis yoga paling populer berdasarkan data pengguna. Endpoint ini hanya mendukung metode HTTP `GET` dan memerlukan autentikasi dengan token Bearer.

Ketika permintaan `GET` diterima, API memeriksa validitas token JWT yang terdapat dalam header `Authorization`. Jika token valid, API mengambil data pengguna dari database dan kemudian menghitung frekuensi setiap jenis yoga dari tabel `History`. Hasilnya diurutkan berdasarkan frekuensi, dan tiga jenis yoga dengan jumlah terbanyak diambil. Persentase popularitas dihitung dari total frekuensi dan dikembalikan dalam format JSON. Jika token tidak valid, API mengembalikan pesan kesalahan dengan status kode 401.

12. Route dan API History

```python
@api.route('/history')
class History_Route(Resource):
    @api.doc(security='Bearer')
    @api.expect(parser4History)
    @api.response(201, 'Created')
    def post(self):
        auth = request.headers.get('Authorization')
        args = parser4History.parse_args()
        waktu = args['waktu']
        jenis_yoga = args['jenis_yoga']
        tanggal = args['tanggal']
        tanggal = tanggal.split("/")
        tanggal = date(int(tanggal[2]),int(tanggal[1]),int(tanggal[0]))
        jwtToken = auth[7:]
        try:
            payload = decode(jwtToken)
            user = db.session.execute(db.select(User).filter_by(id=payload['user_id'])).first()
            user = user[0]
            if not waktu:
                return {
                    'message': 'Waktu tidak boleh kosong'
                }, 400
            if not jenis_yoga:
                return {
                    'message': 'Jenis yoga tidak boleh kosong'
                }, 400
            if not tanggal:
                return {
                    'message': 'Tanggal tidak boleh kosong'
                }, 400
            history = History()
            history.user_id = user.id
            history.tanggal = tanggal
            history.waktu = waktu
            history.jenis_yoga = jenis_yoga
            db.session.add(history)
            db.session.commit()
            return {
                'message' : 'History berhasil ditambahkan'
            }, 201
        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401
    @api.doc(security='Bearer')
    def get(self):
        auth = request.headers.get('Authorization')

        jwtToken = auth[7:]

        try:
            payload = decode(jwtToken)

            histories = db.session.execute(db.select(History).filter_by(user_id=payload['user_id'])).all()
            result = []
            for history in histories:
                history = history._asdict()['History']
                history.tanggal = dumps(history.tanggal, default=str).replace('"','')
                hist = {
                    'id': history.id,
                    'user_id': history.user_id,
                    'tanggal': history.tanggal,
                    'waktu': history.waktu,
                    'jenis_yoga': history.jenis_yoga
                }
                result.append(hist)
        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401

        return result[::-1], 200
```

Gambar 12 Route dan API History

API ini menyediakan endpoint `/history` untuk menambahkan dan mengambil riwayat sesi yoga pengguna. Metode `POST` memvalidasi token JWT dan parameter JSON seperti waktu, jenis yoga, dan tanggal sebelum menambahkan data riwayat baru ke database; jika parameter tidak valid atau token tidak sah, API mengembalikan pesan kesalahan. Metode `GET` memvalidasi token JWT dan mengembalikan riwayat yoga pengguna dari database dalam format JSON, dengan data ditampilkan dalam urutan terbalik; jika token tidak sah, API mengembalikan pesan kesalahan.

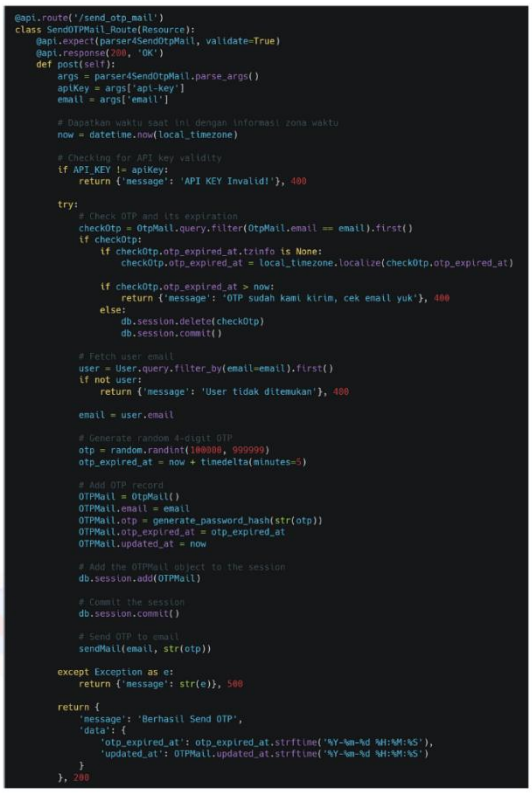
13. Route dan API Feedback

```python
@api.route('/feedback')
class Feedback_Route(Resource):
    @api.doc(security='Bearer')
    @api.expect(parser4Feedback)
    @api.response(201, 'Created')
    def post(self):
        auth = request.headers.get('Authorization')
        args = parser4Feedback.parse_args()
        komentar = args['komentar']

        jwtToken = auth[7:]

        try:
            payload = decode(jwtToken)

            user = db.session.execute(db.select(User).filter_by(id=payload['user_id'])).first()
            user = user[0]

            if not komentar:
                return {
                    'message': 'Komentar tidak boleh kosong'
                }, 400

            feedback = Feedback()
            feedback.user_id = user.id
            feedback.komentar = komentar

            db.session.add(feedback)
            db.session.commit()

            return {
                'message' : 'Feedback berhasil ditambahkan'
            }, 201

        except:
            return {'message': 'Token tidak valid, silahkan masuk dulu mom'}, 401
```

Gambar 13 Route dan API Feedback

API ini menyediakan endpoint `/feedback` untuk mengirimkan umpan balik dari pengguna. Metode `POST` memerlukan autentikasi menggunakan token Bearer, memvalidasi token JWT, dan memproses komentar yang dikirim dalam permintaan JSON. Jika komentar tidak kosong, API menambahkan umpan balik ke database dan mengembalikan pesan sukses dengan status kode 201. Jika komentar kosong atau token tidak valid, API mengembalikan pesan kesalahan yang sesuai dengan status kode 400 atau 401.

14. Route dan API Kirim OTP Email



```python
@api.route('/send_otp_mail')
class SendOTPMail_Route(Resource):
    @api.expect(parser4SendOtpMail, validate=True)
    @api.response(200, 'OK')
    def post(self):
        args = parser4SendOtpMail.parse_args()
        apiKey = args['api-key']
        email = args['email']

        # Dapatkan waktu saat ini dengan informasi zona waktu
        now = datetime.now(local_timezone)

        # Checking for API key validity
        if API_KEY != apiKey:
            return {'message': 'API KEY Invalid!'}, 400

        try:
            # Check OTP and its expiration
            checkOtp = OtpMail.query.filter(OtpMail.email == email).first()
            if checkOtp:
                if checkOtp.otp_expired_at.tzinfo is None:
                    checkOtp.otp_expired_at = local_timezone.localize(checkOtp.otp_expired_at)

                if checkOtp.otp_expired_at > now:
                    return {'message': 'OTP sudah kami kirim, cek email yuk'}, 400
                else:
                    db.session.delete(checkOtp)
                    db.session.commit()

            # Fetch user email
            user = User.query.filter_by(email=email).first()
            if not user:
                return {'message': 'User tidak ditemukan'}, 400

            email = user.email

            # Generate random 4-digit OTP
            otp = random.randint(100000, 999999)
            otp_expired_at = now + timedelta(minutes=5)

            # Add OTP record
            OTPMail = OtpMail()
            OTPMail.email = email
            OTPMail.otp = generate_password_hash(str(otp))
            OTPMail.otp_expired_at = otp_expired_at
            OTPMail.updated_at = now

            # Add the OTPMail object to the session
            db.session.add(OTPMail)

            # Commit the session
            db.session.commit()

            # Send OTP to email
            sendMail(email, str(otp))

        except Exception as e:
            return {'message': str(e)}, 500

        return {
            'message': 'Berhasil Send OTP',
            'data': {
                'otp_expired_at': otp_expired_at.strftime('%Y-%m-%d %H:%M:%S'),
                'updated_at': OTPMail.updated_at.strftime('%Y-%m-%d %H:%M:%S')
            }
        }, 200
```

Gambar 14 Route dan API Kirim OTP Email

API `/send_otp_mail` bertugas untuk mengirimkan OTP (One-Time Password) ke email pengguna. Endpoint ini memerlukan autentikasi API key dan validasi input yang terdiri dari email dan API key. Jika API key tidak valid, akan mengembalikan pesan kesalahan. API ini memeriksa apakah OTP yang sama sudah ada dan belum kedaluwarsa; jika ada, menginformasikan bahwa OTP sudah dikirim. Jika tidak ada OTP aktif atau OTP sudah kedaluwarsa, API akan menghapus catatan OTP yang lama, jika ada, dan menghasilkan OTP baru yang dikirim ke email pengguna. API juga menyimpan informasi OTP baru ke dalam database dan mengirimkan OTP ke email melalui fungsi `sendMail`. Jika ada kesalahan, API mengembalikan pesan kesalahan; jika berhasil, mengembalikan pesan sukses beserta informasi kedaluwarsa OTP.

15. Route dan API Verifikasi OTP Email

```python
@api.route('/verify_otp_mail')
class VerifyOtpMail_Route(Resource):
    @api.expect(parser4VerifyOtpMail, validate=True)
    @api.response(200, 'OK')
    def post(Self):
        args = parser4VerifyOtpMail.parse_args()
        apiKey = args['api-key']
        otp = args['otp']
        email = args['email']

        now = datetime.now(local_timezone)

        #checking for api key is valid
        if API_KEY != apiKey:
            return {'message':'API KEY Invalid!'}, 400

        try:
            #start transaction
            with db.session.begin():

                #check code otp or otp_expired_at < now
                checkOtp = OtpMail.query.filter(OtpMail.email==email).first()

                if checkOtp is None:
                    return {'message': 'Silahkan resend OTP'}, 400

                if checkOtp.otp_expired_at.tzinfo is None:
                    checkOtp.otp_expired_at = local_timezone.localize(checkOtp.otp_expired_at)

                if not check_password_hash(checkOtp.otp, otp):
                    return {'message': 'OTP Invalid'}, 400

                if checkOtp.otp_expired_at < now:
                    return {'message': 'OTP Expired!'}, 400

                #get user by email associated with the OTP
                user = db.session.execute(db.select(User).filter_by(email=checkOtp.email)).first()
                if not user:
                    return {'message': 'Akun tidak ditemukan!'}, 400

                # Remove the OTP record after successful verification
                db.session.delete(checkOtp)


        except Exception as e:
            #rollback here
            return {'message': str(e)}, 500

        return {
            'message' : 'Berhasil Verifikasi OTP'
        }, 200
```

Gambar 15 Route dan API Verifikasi OTP Email

API `/verify_otp_mail` digunakan untuk memverifikasi OTP (One-Time Password) yang dikirim ke email pengguna. Endpoint ini memerlukan API key, OTP, dan email sebagai input. Pertama, API memeriksa kevalidan API key; jika tidak valid, mengembalikan pesan kesalahan. API kemudian memulai transaksi database dan memeriksa apakah OTP yang diberikan sesuai dengan OTP yang tersimpan dan belum kedaluwarsa. Jika OTP valid dan belum kedaluwarsa, API akan mencari pengguna berdasarkan email terkait OTP tersebut, menghapus catatan OTP dari database setelah verifikasi berhasil, dan mengembalikan pesan sukses. Jika terjadi kesalahan, API akan membatalkan transaksi dan mengembalikan pesan kesalahan yang sesuai.

16. Fungsi, Route, dan API Deteksi Gerakan Senam Yoga Hamil

```
def point():
    x = [
        'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20', 'x21', 'x22',
        'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33',
    ]
    y = [
        'y12', 'y13', 'y14', 'y15', 'y16', 'y17', 'y18', 'y19', 'y20', 'y21', 'y22',
        'y23', 'y24', 'y25', 'y26', 'y27', 'y28', 'y29', 'y30', 'y31', 'y32', 'y33',
    ]
    z = [
        'z12', 'z13', 'z14', 'z15', 'z16', 'z17', 'z18', 'z19', 'z20', 'z21', 'z22',
        'z23', 'z24', 'z25', 'z26', 'z27', 'z28', 'z29', 'z30', 'z31', 'z32', 'z33',
    ]
    v = [
        'v12', 'v13', 'v14', 'v15', 'v16', 'v17', 'v18', 'v19', 'v20', 'v21', 'v22',
        'v23', 'v24', 'v25', 'v26', 'v27', 'v28', 'v29', 'v30', 'v31', 'v32', 'v33',
    ]
    coords = [x, y, z, v]
    return coords
```

Gambar 16 Fungsi Point

Fungsi point diatas mengembalikan daftar koordinat yang dinamakan x, y, z, dan v. Setiap daftar berisi string yang mewakili titik-titik tertentu, mungkin untuk tujuan pelacakan atau anotasi di suatu model. Fungsi ini hanya mengembalikan daftar koordinat dalam bentuk list yang berisi list lainnya, masing-masing untuk variabel x, y, z, dan v.

```
def base64_to_image(base64_string):
    # Extract the base64 encoded binary data from the input string
    base64_data = base64_string.split(",")[1]
    # Decode the base64 data to bytes
    image_bytes = base64.b64decode(base64_data)
    # Convert the bytes to numpy array
    image_array = np.frombuffer(image_bytes, dtype=np.uint8)
    # Decode the numpy array as an image using OpenCV
    image = cv2.imdecode(image_array, cv2.IMREAD_COLOR)
    return image
```

Gambar 17 Fungsi Base 64 To Image

Fungsi base 64 to image menerima string base64 yang mewakili gambar dan mengubahnya menjadi format gambar yang dapat diproses. Pertama, ia mengekstrak data base64 dari string input, kemudian mendekode data tersebut menjadi bytes. Bytes tersebut diubah menjadi array numpy dan akhirnya didekode menjadi gambar menggunakan OpenCV. Hasil akhirnya adalah gambar dalam format OpenCV.

```python
@socketio.on('connect')
def handle_connect():
    print('Client connected')
    emit("my response", {"data": "Connected"})

@socketio.on('disconnect')
def handle_disconnect():
    print('Client disconnected')

@socketio.on_error()
def handle_error(e):
    print('An error occurred:', str(e))

@socketio.on('message')
def handle_message(data):
    print('Received message:', data)
    emit('response', 'Server received message: ' + data)
```

Gambar 18 Socket

Handlers untuk WebSocket dengan `socketio` meliputi beberapa fungsi: `handle_connect()` yang menangani koneksi klien ke server WebSocket dengan mencetak pesan ke konsol dan mengirimkan balasan bahwa koneksi berhasil; `handle_disconnect()` yang menangani pemutusan koneksi klien dengan mencetak pesan ke konsol saat klien terputus; `handle_error(e)` yang menangani kesalahan selama interaksi WebSocket dengan mencetak pesan kesalahan ke konsol; dan `handle_message(data)` yang menangani pesan dari klien dengan mencetak pesan yang diterima dan mengirimkan balasan yang menyatakan bahwa pesan telah diterima.

```
@socketio.on("image")
def handle_image(imageData):
    try:
        image_data_bytes = base64.b64decode(imageData['imageData'])
        image_array = np.frombuffer(image_data_bytes, dtype=np.uint8)
        decoded_image = cv2.imdecode(image_array, cv2.IMREAD_UNCHANGED)
        mp_drawing = mp.solutions.drawing_utils # Drawing helpers.
        mp_holistic = mp.solutions.holistic     # Mediapipe Solutions.
        with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
            image = cv2.cvtColor(decoded_image, cv2.COLOR_BGR2RGB)
            image = cv2.resize(image, (340, 180), interpolation=cv2.INTER_LINEAR)
            results = holistic.process(image)
            image.flags.writeable = True
            image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
            # Pose Detections
            mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS,
                                      mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=4),
                                      mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                                    )
            # Export coordinates
            try:
            # Extract Pose landmarks
                pose = results.pose_landmarks.landmark
                pose_row = list(np.array([[landmark.x, landmark.y, landmark.z, landmark.visibility] for landmark in pose]).flatten())
                # Concate rows
                row = pose_row
                # point[11] - point[32] input to tflite model.
                coords = point()
                specify_float = 8
                dict_p12_to_p33 = {
                    # x12 to x33
                    coords[0][0]:round(row[44], specify_float),
                    coords[0][1]:round(row[48], specify_float),
                    coords[0][2]:round(row[52], specify_float),
                    coords[0][3]:round(row[56], specify_float),
                    coords[0][4]:round(row[60], specify_float),
                    coords[0][5]:round(row[64], specify_float),
                    coords[0][6]:round(row[68], specify_float),
                    coords[0][7]:round(row[72], specify_float),
                    coords[0][8]:round(row[76], specify_float),
                    coords[0][9]:round(row[80], specify_float),
                    coords[0][10]:round(row[84], specify_float),
                    coords[0][11]:round(row[88], specify_float),
                    coords[0][12]:round(row[92], specify_float),
                    coords[0][13]:round(row[96], specify_float),
                    coords[0][14]:round(row[100], specify_float),
                    coords[0][15]:round(row[104], specify_float),
                    coords[0][16]:round(row[108], specify_float),
                    coords[0][17]:round(row[112], specify_float),
                    coords[0][18]:round(row[116], specify_float),
                    coords[0][19]:round(row[120], specify_float),
                    coords[0][20]:round(row[124], specify_float),
                    coords[0][21]:round(row[128], specify_float),
                    # y12 to y33
                    coords[1][0]:round(row[45], specify_float),
                    coords[1][1]:round(row[49], specify_float),
                    coords[1][2]:round(row[53], specify_float),
                    coords[1][3]:round(row[57], specify_float),
                    coords[1][4]:round(row[61], specify_float),
                    coords[1][5]:round(row[65], specify_float),
                    coords[1][6]:round(row[69], specify_float),
                    coords[1][7]:round(row[73], specify_float),
                    coords[1][8]:round(row[77], specify_float),
                    coords[1][9]:round(row[81], specify_float),
                    coords[1][10]:round(row[85], specify_float),
                    coords[1][11]:round(row[89], specify_float),
                    coords[1][12]:round(row[93], specify_float),
                    coords[1][13]:round(row[97], specify_float),
                    coords[1][14]:round(row[101], specify_float),
                    coords[1][15]:round(row[105], specify_float),
                    coords[1][16]:round(row[109], specify_float),
                    coords[1][17]:round(row[113], specify_float),
                    coords[1][18]:round(row[117], specify_float),
                    coords[1][19]:round(row[121], specify_float),
                    coords[1][20]:round(row[125], specify_float),
                    coords[1][21]:round(row[129], specify_float),
```

```
                    # z12 to z33
                    coords[2][0]:round(row[46], specify_float),
                    coords[2][1]:round(row[50], specify_float),
                    coords[2][2]:round(row[54], specify_float),
                    coords[2][3]:round(row[58], specify_float),
                    coords[2][4]:round(row[62], specify_float),
                    coords[2][5]:round(row[66], specify_float),
                    coords[2][6]:round(row[70], specify_float),
                    coords[2][7]:round(row[74], specify_float),
                    coords[2][8]:round(row[78], specify_float),
                    coords[2][9]:round(row[82], specify_float),
                    coords[2][10]:round(row[86], specify_float),
                    coords[2][11]:round(row[90], specify_float),
                    coords[2][12]:round(row[94], specify_float),
                    coords[2][13]:round(row[98], specify_float),
                    coords[2][14]:round(row[102], specify_float),
                    coords[2][15]:round(row[106], specify_float),
                    coords[2][16]:round(row[110], specify_float),
                    coords[2][17]:round(row[114], specify_float),
                    coords[2][18]:round(row[118], specify_float),
                    coords[2][19]:round(row[122], specify_float),
                    coords[2][20]:round(row[126], specify_float),
                    coords[2][21]:round(row[130], specify_float),
                    # v12 to v33
                    coords[3][0]:round(row[47], specify_float),
                    coords[3][1]:round(row[51], specify_float),
                    coords[3][2]:round(row[55], specify_float),
                    coords[3][3]:round(row[59], specify_float),
                    coords[3][4]:round(row[63], specify_float),
                    coords[3][5]:round(row[67], specify_float),
                    coords[3][6]:round(row[71], specify_float),
                    coords[3][7]:round(row[75], specify_float),
                    coords[3][8]:round(row[79], specify_float),
                    coords[3][9]:round(row[83], specify_float),
                    coords[3][10]:round(row[87], specify_float),
                    coords[3][11]:round(row[91], specify_float),
                    coords[3][12]:round(row[95], specify_float),
                    coords[3][13]:round(row[99], specify_float),
                    coords[3][14]:round(row[103], specify_float),
                    coords[3][15]:round(row[107], specify_float),
                    coords[3][16]:round(row[111], specify_float),
                    coords[3][17]:round(row[115], specify_float),
                    coords[3][18]:round(row[119], specify_float),
                    coords[3][19]:round(row[123], specify_float),
                    coords[3][20]:round(row[127], specify_float),
                    coords[3][21]:round(row[131], specify_float),
                }
                #input coordinat to predict
                input_dict = {name: np.expand_dims(np.array(value, dtype=np.float32), axis=0) for name, value in
dict_p12_to_p33.items()}
                # Make Detections
                model = tf.keras.models.load_model('./model/model_new.h5')
                result = model.predict(input_dict)
                result = result[0]
                body_language_class = np.argmax(result)
                body_language_prob = result[np.argmax(result)]
                # 1: cat-cow-pose, 2: child-pose, 4: lateral-leg-raise, 5: side-bend, 6: sideclamp , 7: savasana
                if str(body_language_class) == '1':
                    pose_class = 'Cat Cow Pose'
                elif str(body_language_class) == '2':
                    pose_class = 'Child Pose'
                elif str(body_language_class) == '4':
                    pose_class = 'Lateral Leg Raise Pose'
                elif str(body_language_class) == '5':
                    pose_class = 'Side Bend Pose'
                elif str(body_language_class) == '6':
                    pose_class = 'Side Clamp Pose'
                elif str(body_language_class) == '7':
                    pose_class = 'Savasana Pose'
                else:
                    pose_class = 'GAK KEDETEKSI'
                print(f'class: {body_language_class}, prob: {body_language_prob}')
                # Show pose category near the ear.
                coords = tuple(np.multiply(
                    np.array(
                        (results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_EAR].x,
                        results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_EAR].y)),
                    [1280,480]
                ).astype(int))
            except:
                pass
        processed_image_bytes = cv2.imencode('.jpg', image)[1].tobytes()
        processed_image_data = base64.b64encode(processed_image_bytes).decode('utf-8')
        prob = str(body_language_prob)
        emit('imageResponse', {"imageData": processed_image_data,"pose_class": pose_class, "prob": prob})
    except Exception as e:
        print('Error processing image:', e)
```

Fungsi ini mengolah gambar yang dikirim oleh klien dalam format base64. Pertama, gambar di-decode dari format base64 menjadi bytes, lalu diubah menjadi array numpy, dan didekode menggunakan OpenCV.

Kemudian, dengan menggunakan MediaPipe, gambar diproses untuk mendeteksi landmark pose. Data landmark pose digunakan untuk ekstraksi fitur dan diproses untuk prediksi dengan model TensorFlow yang sudah dilatih. Hasil prediksi menunjukkan kelas pose yoga berdasarkan landmark pose. Nama kelas ditetapkan berdasarkan hasil prediksi, dan informasi tentang pose yoga dipilih untuk ditampilkan.

Gambar yang telah diproses kemudian dikodekan kembali menjadi format base64 dan dikirimkan kembali ke klien bersama dengan informasi tentang kelas pose dan probabilitas prediksi. Jika terjadi kesalahan selama proses, error dicetak ke konsol.

REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

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	:	EC00202471215, 25 Juli 2024
Pencipta		
Nama	:	**Maulana Alamsyah, Hepatika Zidny Ilmadina dkk**
Alamat	:	JL. KH. Umar Asnawi Desa Kebasen RT 08 RW 02 Kec.Talang Kab.Tegal Provinsi Jawa Tengah 52193, Talang, Tegal, Jawa Tengah, 52193
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**
Alamat	:	Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**Pengembangan Aplikasi PREG-FIT Berbasis Mobile Untuk Membantu Ibu Hamil Melakukan Senam Yoga Prenatal**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	25 Juli 2024, di Tegal
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000646566

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

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

IGNATIUS M.T. SILALAHI
NIP. 196812301996031001

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

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Maulana Alamsyah	JL. KH. Umar Asnawi Desa Kebasen RT 08 RW 02 Kec.Talang Kab.Tegal Provinsi Jawa Tengah 52193, Talang, Tegal
2	Hepatika Zidny Ilmadina	Jalan Kenanga Gang 1 Nomor 9, Kelurahan Mangkukusuman, Kecamatan Tegal Timur Kota Tegal, Provinsi Jawa Tengah 52123, Tegal Timur, Tegal
3	Dwi Intan Af'idah	Desa Grinting RT003/RW001, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah 52253, Bulakamba, Brebes

SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN SKRIPSI

Nama : **Maulana Alamsyah**
Nim : **20090137**
No. Ponsel : **089636092727**
Judul TA : **Pengembangan Aplikasi Preg-Fit Berbasis Mobile Untuk Membantu Ibu Hamil Melakukan Senam Yoga Prenatal**
Dosen Pembimbing I : **Hepatika Zidny Ilmadina, S.Pd., M.Kom.**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21 Maret 2024	Target Luaran	Pastikan hasil luaran hasil skripsi	
2.	19 April 2024	Aplikasi	Flow chatbot dan reminder	
3.	17 Mei 2024	Aplikasi	Implementasi chatbot dan reminder	
4.	20 Mei 2024	Aplikasi	Pemodelan Chatbot	
5.	04 Juli 2024	Aplikasi	Integrasi Model dengan Aplikasi Preg-Fit	
6.	22 Juli 2024	Aplikasi, HKI	Perbaikan dataset chatbot	
7.	24 Juli 2024	Laporan	Revisi Laporan	

SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
8.	25 Juli 2024	Laporan	ACC Laporan	

Tegal, 26 Juli 2024
Dosen Pembimbing I

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

SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN SKRIPSI

Nama : **Maulana Alamsyah**
Nim : **20090137**
No. Ponsel : **089636092727**
Judul TA : **Pengembangan Aplikasi Preg-Fit Berbasis Mobile Untuk Membantu Ibu Hamil Melakukan Senam Yoga Prenatal**
Dosen Pembimbing I : **Dwi Intan Af'idah, S.T., M.Kom.**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21 Maret 2024	Target Luaran	Pastikan hasil luaran hasil skripsi	
2.	19 April 2024	Aplikasi	Segera menyelesaikan fitur chatbot dan reminder	
3.	17 Mei 2024	Aplikasi	Segera Implementasi chatbot dan reminder	
4.	20 Mei 2024	Aplikasi	Dilanjutkan Pemodelan Chatbot yang lebih optimal	
5.	04 Juli 2024	Aplikasi	Integrasi Model dengan Aplikasi Preg-Fit	
6.	22 Juli 2024	Aplikasi, HKI	Perbaikan dataset chatbot	

SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
7.	24 Juli 2024	Laporan	Revisi Laporan	
8.	25 Juli 2024	Laporan	ACC Laporan	

Tegal, 26 Juli 2024
Dosen Pembimbing II

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