

LAMPIRAN

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Desty Nurul Anitsa
NIM : 20090134
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dyah Apriliani, S.T., M.Kom.
Status : Dosen Tetap
NIDN : 0614049002
Jabatan Fungsional : Lektor
Pangkat/Golongan : III D

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



Desty Nurul Anitsa

Pihak Kedua



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

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Desty Nurul Anitsa
NIM : 20090134
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 II Skripsi Pihak Pertama dengan **syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua.** Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 25 Maret 2024

Pihak Pertama



Desty Nurul Anitsa

Pihak Kedua



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

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Desty Nurul Anitsa
Kewarganegaraan : Indonesia
Alamat : Desa Pegirikan RT 32 RW 08, Kecamatan Talang,
Kabupaten Tegal.
2. Nama : Dyah Apriliani. S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68, Randugunting, Kecamatan Tegal Selatan,
Kota Tegal
3. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jalan kenanga Gang 1 Nomor 9, Kelurahan Mangkukusuman,
Kecamatan Tegal Timur Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Aplikasi Mobile Android
Berjudul : Pengembangan aplikasi *Mobile Bi-Cara* menggunakan algoritma *Long Short Term Memory* dalam berkomunikasi dengan bahasa isyarat
 - 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 pidana 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 RI dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditanggungkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 19 Juli 2024



(Desty Nurul Anissa)
Pemegang Hak Cipta *

(Dyah Aprilia S.P., M.Kom.)
Pemegang Hak Cipta *

(Hepatika Ziday Ilmudina, S.Pd., 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 : Deity Nurul Anissa
Kewarganegaraan : Indonesia
Alamat : Desa Pegirilon RT 32 RW 08, Kecamatan Talang,
Kabupaten Tegal.
2. Nama : Dyah Apriliani. S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Block C4, Jl. Nangka Gg. 2 RT 02
RW 02, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal
3. Nama : Hepatika Zidny Ilmadina, S.Pd., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jalan keanaga Gang 1 Nomor 9, Kelurahan Mangkukusuman,
Kecamatan Tegal Timur Kota Tegal

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

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

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Pengembangan aplikasi *Mobile Bi-Caru* menggunakan algoritma *Long Short Term Memory* dalam berkomunikasi dengan bahasa isyarat". untuk didaftarkan di Direktorat Hak Cipta dan Domain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.


Tegal, 19 Juli 2024




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



(Deity Nurul Anissa)

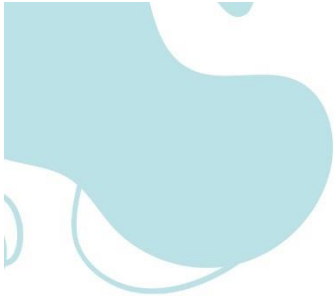


(Dyah Apriliani, S.T., M.Kom.)



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

Manual Book



Manual Book

Aplikasi Bi-Cara



Oleh :

Desty Nurul Anitsa

Dyah Apriliani, S.T., M.Kom

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



Bi-Cara



Pendahuluan

1. Latar Belakang

Komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara adalah sebuah tantangan penting dalam lingkungan sosial dan kesehatan. Sebagian besar dari mereka bergantung pada bahasa isyarat, seperti Sistem Isyarat Bahasa Indonesia (SIBI), untuk berinteraksi dengan lingkungan sekitar. Meskipun SIBI diakui sebagai bahasa isyarat resmi di Indonesia, masih banyak individu yang tidak memahaminya, termasuk tenaga kesehatan di fasilitas medis. Hal ini seringkali menghambat komunikasi yang efektif dan mengakibatkan kesulitan dalam mendapatkan layanan kesehatan yang sesuai.

Data yang dikumpulkan dari observasi menunjukkan bahwa sebagian besar fasilitas kesehatan, seperti rumah sakit dan klinik, belum sepenuhnya ramah terhadap kebutuhan komunikasi penyandang disabilitas. Dalam sebuah penelitian di Rumah Sakit Mitra Siaga Tegal, sekitar 41 dari 47 tenaga kesehatan yang disurvei mengakui bahwa mereka tidak dapat menggunakan bahasa isyarat. Kurangnya pemahaman ini berdampak pada kualitas interaksi antara tenaga kesehatan dan pasien tunarungu maupun tunawicara, yang seringkali mengalami kesulitan dalam menjelaskan keluhan atau memahami informasi medis yang diberikan.

Bi-Cara hadir sebagai solusi yang inovatif untuk mengatasi tantangan ini. Aplikasi ini menggunakan teknologi terbaru dalam pengenalan bahasa isyarat dan kecerdasan buatan berbasis LSTM. Dengan Bi-Cara, pengguna dapat dengan mudah mengonversi gestur tangan dalam bahasa isyarat SIBI menjadi suara yang dapat dipahami oleh individu normal atau tenaga kesehatan yang tidak menguasai bahasa isyarat. Hal ini tidak hanya meningkatkan aksesibilitas komunikasi, tetapi juga mempromosikan inklusi sosial bagi penyandang disabilitas dalam lingkungan pelayanan kesehatan.



2. Tujuan Pembuatan Dokumen

Dokumen manual aplikasi Bi-Cara ini disusun dengan tujuan untuk memberikan panduan yang jelas dan mudah dipahami bagi pengguna dalam menggunakan aplikasi Bi-Cara. Dokumen ini berisi informasi lengkap mengenai deskripsi aplikasi, cara penggunaan, dan fitur-fitur yang tersedia. Diharapkan dengan tersedianya dokumen ini, pengguna dapat dengan lebih mudah memahami dan memanfaatkan aplikasi Bi-Cara untuk meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara di lingkungan pelayanan kesehatan. Dokumen ini juga merupakan bagian dari upaya untuk mencapai tujuan utama pengembangan aplikasi Bi-Cara, yaitu memfasilitasi komunikasi yang lebih efektif dan inklusif dalam masyarakat.

3. Nilai Inovasi

Aplikasi Bi-Cara menawarkan nilai inovasi dalam meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu maupun tunawicara. Dengan menggunakan teknologi pengenalan bahasa isyarat SIBI dan kecerdasan buatan berbasis LSTM, Bi-Cara memungkinkan pengguna untuk mengonversi gestur tangan dalam bahasa isyarat menjadi suara yang dapat dipahami oleh orang-orang yang tidak mengerti bahasa isyarat. Hal ini tidak hanya memfasilitasi komunikasi yang lebih efektif di lingkungan pelayanan kesehatan, tetapi juga mempromosikan inklusi sosial dengan memberdayakan individu disabilitas untuk berpartisipasi secara aktif dalam berbagai aspek kehidupan.



4. Dampak Pemanfaatan Aplikasi Bi-Cara

- Meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara di lingkungan pelayanan kesehatan.
- Memfasilitasi interaksi yang lebih baik antara penyandang disabilitas dan tenaga kesehatan, mengurangi kesenjangan komunikasi.
- Memberdayakan individu disabilitas untuk berpartisipasi aktif dalam masyarakat dengan meningkatkan kemampuan komunikasi mereka.
- Mengurangi ketergantungan pada penerjemah bahasa isyarat yang tidak selalu tersedia di setiap fasilitas kesehatan.
- Mendorong inklusi sosial dengan mempromosikan pemahaman dan dukungan terhadap keberagaman komunikasi.
- Menawarkan solusi teknologi yang praktis dan mudah diakses, meningkatkan kualitas hidup individu disabilitas melalui aksesibilitas komunikasi yang lebih baik.



Pembahasan Aplikasi

1. Perangkat yang dibutuhkan

Perangkat Lunak dan Perangkat Keras

- Perangkat lunak yang digunakan adalah *android operating system*
- Perangkat keras yang digunakan hanya smartphone

2. Deskripsi Fungsional Aplikasi Bi-Cara

Tujuan Utama

Aplikasi ini dapat menjadi solusi untuk membantu individu dengan disabilitas tunarungu dan tunawicara berkomunikasi dengan lebih efektif, terutama di lingkungan pelayanan kesehatan. Aplikasi Bi-Cara menyediakan alat yang praktis dan mudah diakses untuk mengonversi bahasa isyarat SIBI menjadi teks atau suara, sehingga mempermudah interaksi dengan tenaga kesehatan maupun masyarakat umum yang tidak menguasai bahasa isyarat.

Menu

1. Home
2. Profil



3. Penjelasan Detail Fitur

Semua fitur pada Aplikasi Bi-Cara hanya dapat digunakan apabila terhubung dengan internet, Terdapat beberapa fitur pada aplikasi Bi-Cara diantaranya :

- **Fitur Daftar**

Fitur ini memungkinkan pengguna baru untuk membuat akun pada aplikasi Bi-Cara. Pengguna akan diminta mengisi informasi dasar seperti nama, jenis kelamin, alamat email, dan kata sandi. Setelah mengisi formulir pendaftaran, pengguna akan menerima kode verifikasi OTP yang dikirimkan ke email. Kemudian pengguna dapat memasukkan kode OTP untuk diverifikasi.

- **Fitur Login** Fitur login memungkinkan pengguna yang sudah terdaftar untuk mengakses akun mereka dengan memasukkan alamat email dan kata sandi. Fitur ini juga dilengkapi dengan opsi "Lupa Password" untuk membantu pengguna memulihkan akses akun mereka jika lupa kata sandi.

- **Fitur Verifikasi Menggunakan Kode OTP**

Fitur ini memungkinkan pengguna untuk memverifikasi identitas mereka menggunakan kode OTP (*One-Time Password*) yang dikirimkan ke alamat email yang didaftarkan. Pengguna harus memasukkan kode OTP yang diterima untuk melanjutkan proses verifikasi. Fitur ini penting untuk memastikan keamanan tambahan dan validasi identitas pengguna.

- **Fitur Ubah Profil**

Fitur ini memungkinkan pengguna untuk memperbarui informasi profil mereka, seperti nama dan jenis kelamin. Pengguna dapat mengakses fitur ini dari menu profil dan melakukan perubahan yang diperlukan.





Bi-Cara

- **Fitur Deteksi Gerakan Bahasa Isyarat ke Suara**
Fitur utama dari aplikasi Bi-Cara adalah kemampuan untuk mendeteksi gerakan bahasa isyarat SIBI dan mengonversinya menjadi suara. Fitur ini menggunakan teknologi pengenalan gerakan tangan dan kecerdasan buatan untuk menganalisis gestur bahasa isyarat dan menghasilkan output suara yang sesuai. Ini membantu penyandang disabilitas tunarungu dan tunawicara untuk berkomunikasi dengan lebih mudah dengan orang-orang yang tidak memahami bahasa isyarat.
- **Fitur Feedback**
Fitur ini memungkinkan pengguna untuk memberikan umpan balik tentang pengalaman mereka menggunakan aplikasi Bi-Cara. Pengguna dapat menyampaikan saran, kritik, atau melaporkan masalah yang mereka temui. Fitur ini penting untuk pengembangan aplikasi lebih lanjut dan memastikan bahwa kebutuhan pengguna terpenuhi dengan baik.
- **Fitur Keluar/Logout**
Fitur ini memungkinkan pengguna untuk keluar dari akun mereka dengan aman setelah selesai menggunakan aplikasi. Pengguna dapat mengakses opsi logout dari menu profil, memastikan bahwa sesi mereka berakhir dan akun mereka tetap aman dari akses tidak sah.



Penggunaan Aplikasi

4. Penggunaan Aplikasi Bi-Cara

a. Cara Install Aplikasi Bi-Cara

- Aplikasi Bi-Cara belum terdaftar pada Play Store sehingga masih menggunakan file .apk, lakukan penginstalan seperti menginstall aplikasi pada umumnya.
Setelah aplikasi berhasil terpasang, kemudian klik buka dan akan masuk pada aplikasi.

1107 0 0 0 1107 0 0 0

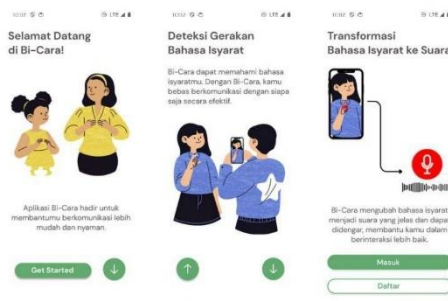


Gambar 1 Halaman Splashscreen aplikasi

b. Cara Pengguna Daftar Dalam Aplikasi Bi-Cara

Berikut ini merupakan cara pengguna daftar dalam aplikasi Bi-Cara:

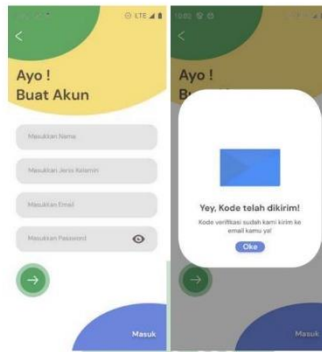
- Pastikan smartphone terhubung dengan internet, kemudian buka Aplikasi Bi-Cara.
- Setelah membuka aplikasi, akan masuk ke halaman splash screen, kemudian pengguna akan diarahkan ke halaman onboarding, halaman ini menampilkan informasi singkat mengenai Aplikasi Bi-Cara. Halaman ke tiga dari onboarding, terdapat tombol untuk login dan daftar. Untuk mendaftar aplikasi pengguna dapat memilih tombol "Daftar"



Gambar 2 Halaman Onboarding

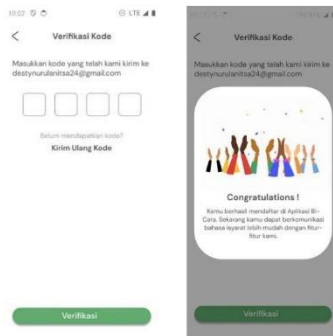
Bi-Cara

- Pengguna memasukkan nama, jenis kelamin, alamat email dan password lalu pengguna mengklik lanjut. Pengguna akan diarahkan ke halaman verifikasi kode, Pengguna memasukkan kode verifikasi OTP sesuai dengan yang didapat pada email yang didaftarkan.



Gambar 3 Halaman Daftar

- Langkah selanjutnya adalah verifikasi OTP yang dikirimkan ke alamat email, pengguna dapat memasukkan kode OTP yang didapat.



Gambar 4 Halaman Verifikasi Akun

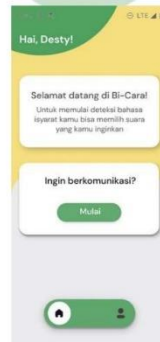
c. Cara Pengguna Masuk Dalam Aplikasi Bi-Cara

- Setelah berhasil mendaftar, pengguna dapat membuka aplikasi, dihalaman ke tiga onboarding, pengguna dapat memilih tombol "Masuk". Kemudian pengguna akan diarahkan pada halaman masukan email dan password dan klik lanjut.



Gambar 5 Halaman Masuk

- Setelah masuk, pengguna akan diarahkan ke halaman home.



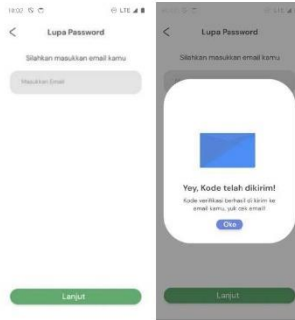
Gambar 6 Halaman Home.

d. Cara Pengguna Lupa Password Aplikasi Bi-Cara

- Pada halaman masuk, pengguna dapat mengklik Lupa Password.
- Pengguna dapat memasukkan alamat email, lalu menekan lanjut untuk mendapatkan kode verifikasi.



Bi-Cara



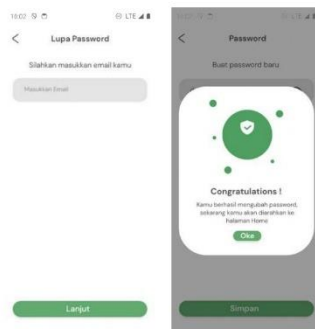
Gambar 7 Halaman Lupa Password

- Pengguna memasukan kode sesuai dengan yang dikirim ke alamat email.



Gambar 8 Halaman Verifikasi.

- Selanjutnya penggunaan diarahkan untuk memasukan password baru lalu menekan lanjut untuk mengubah password.



Gambar 9 Halaman Buat Passsword Baru

Bi-Cara

d. Cara Pengguna Ubah Profil

- Di halaman home pengguna menekan menu profil.



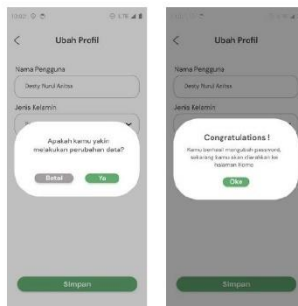
Gambar 10 Halaman Profil

- Pada halaman profil, pengguna dapat mengedit atau mengubah data diri menggunakan tombol edit.



Gambar 11 Halaman Edit Profil

- Pengguna mengubah data diri di profil, menekan simpan. Pengguna mengkonfirmasi perubahan dengan menekan Ya. Maka, Perubahan profil diterapkan.

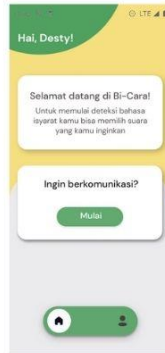


Gambar 12 Pop up Konfirmasi Edit Profil

Bi-Cara

e. Cara Melakukan Komunikasi Bahasa SIBI Aplikasi Bi-Cara

- Di halaman home pengguna menekan tombol mulai.



Gambar 13 Halaman Home

- pengguna menekan ikon kamera untuk membalikan kamera, ikon video untuk memulai komunikasi.



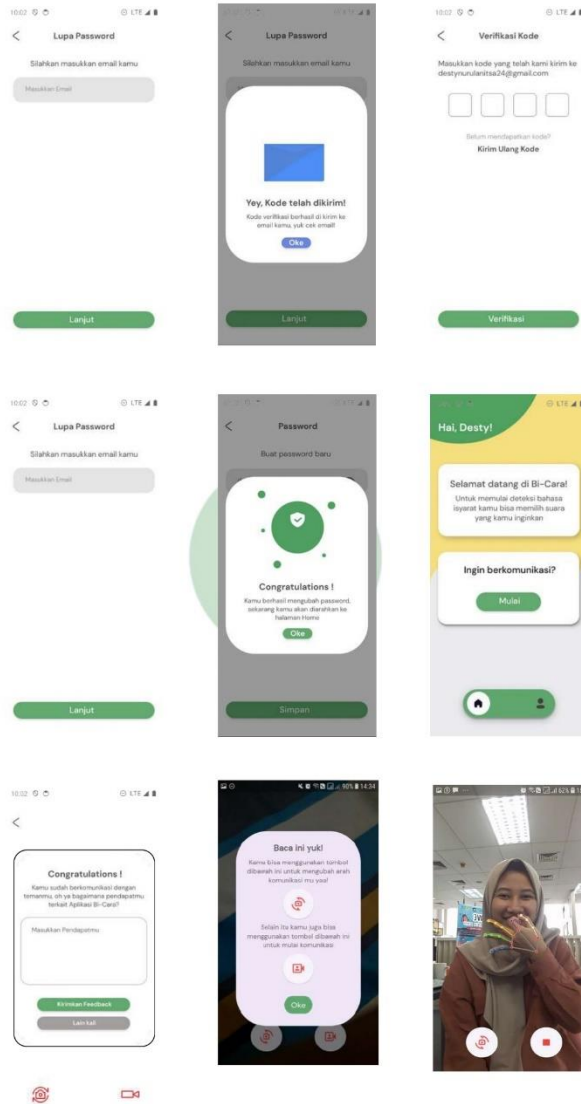
Gambar 14 Gambar Informasi Penggunaan Fitur Komunikasi

- Pengguna berkomunikasi bahasa SIBI dengan mengarahkan kamera dan melakukan gerakan bahasa isyarat, suara hasil deteksi akan berbunyi ketika gerakan sudah selesai.

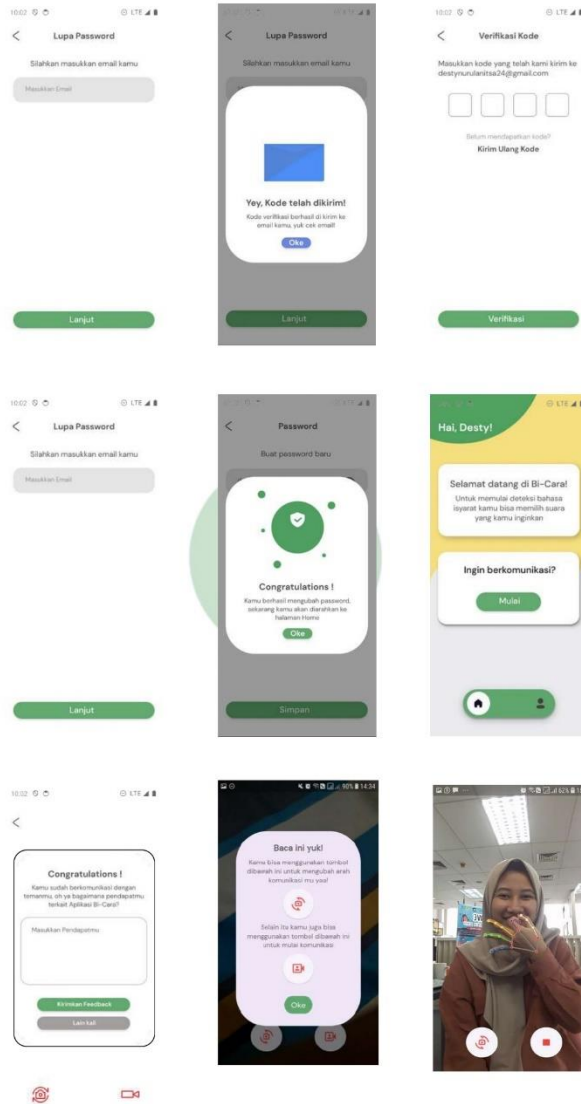


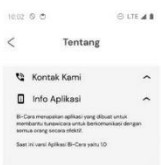
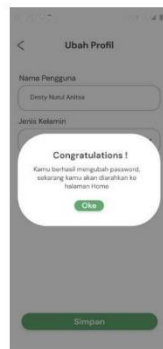
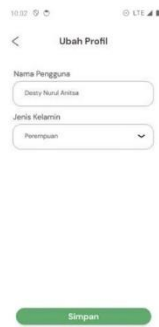
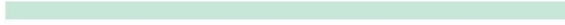
Gambar 15 Fitur Komunikasi

Bi-Cara



Bi-Cara





Dokumen Teknikal



Dokumen Teknis

Aplikasi Bi-Cara



Oleh :

Desty Nurul Anitsa

Dyah Apriliani, S.T., M.Kom

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



Bi-Cara



Pendahuluan

1. Latar Belakang

Komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara adalah sebuah tantangan penting dalam lingkungan sosial dan kesehatan. Sebagian besar dari mereka bergantung pada bahasa isyarat, seperti Sistem Isyarat Bahasa Indonesia (SIBI), untuk berinteraksi dengan lingkungan sekitar. Meskipun SIBI diakui sebagai bahasa isyarat resmi di Indonesia, masih banyak individu yang tidak memahaminya, termasuk tenaga kesehatan di fasilitas medis. Hal ini seringkali menghambat komunikasi yang efektif dan mengakibatkan kesulitan dalam mendapatkan layanan kesehatan yang sesuai.

Data yang dikumpulkan dari observasi menunjukkan bahwa sebagian besar fasilitas kesehatan, seperti rumah sakit dan klinik, belum sepenuhnya ramah terhadap kebutuhan komunikasi penyandang disabilitas. Dalam sebuah penelitian di Rumah Sakit Mitra Siaga Tegal, sekitar 41 dari 47 tenaga kesehatan yang disurvei mengakui bahwa mereka tidak dapat menggunakan bahasa isyarat. Kurangnya pemahaman ini berdampak pada kualitas interaksi antara tenaga kesehatan dan pasien tunarungu maupun tunawicara, yang seringkali mengalami kesulitan dalam menjelaskan keluhan atau memahami informasi medis yang diberikan.

Bi-Cara hadir sebagai solusi yang inovatif untuk mengatasi tantangan ini. Aplikasi ini menggunakan teknologi terbaru dalam pengenalan bahasa isyarat dan kecerdasan buatan berbasis LSTM. Dengan Bi-Cara, pengguna dapat dengan mudah mengonversi gestur tangan dalam bahasa isyarat SIBI menjadi suara yang dapat dipahami oleh individu normal atau tenaga kesehatan yang tidak menguasai bahasa isyarat. Hal ini tidak hanya meningkatkan aksesibilitas komunikasi, tetapi juga mempromosikan inklusi sosial bagi penyandang disabilitas dalam lingkungan pelayanan kesehatan.



2. Tujuan Pembuatan Dokumen

Dokumen manual aplikasi Bi-Cara ini disusun dengan tujuan untuk memberikan panduan yang jelas dan mudah dipahami bagi pengguna dalam menggunakan aplikasi Bi-Cara. Dokumen ini berisi informasi lengkap mengenai deskripsi aplikasi, cara penggunaan, dan fitur-fitur yang tersedia. Diharapkan dengan tersedianya dokumen ini, pengguna dapat dengan lebih mudah memahami dan memanfaatkan aplikasi Bi-Cara untuk meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara di lingkungan pelayanan kesehatan. Dokumen ini juga merupakan bagian dari upaya untuk mencapai tujuan utama pengembangan aplikasi Bi-Cara, yaitu memfasilitasi komunikasi yang lebih efektif dan inklusif dalam masyarakat.

3. Nilai Inovasi

Aplikasi Bi-Cara menawarkan nilai inovasi dalam meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu maupun tunawicara. Dengan menggunakan teknologi pengenalan bahasa isyarat SIBI dan kecerdasan buatan berbasis LSTM, Bi-Cara memungkinkan pengguna untuk mengonversi gestur tangan dalam bahasa isyarat menjadi suara yang dapat dipahami oleh orang-orang yang tidak mengerti bahasa isyarat. Hal ini tidak hanya memfasilitasi komunikasi yang lebih efektif di lingkungan pelayanan kesehatan, tetapi juga mempromosikan inklusi sosial dengan memberdayakan individu disabilitas untuk berpartisipasi secara aktif dalam berbagai aspek kehidupan.



4. Dampak Pemanfaatan Aplikasi Bi-Cara

- Meningkatkan aksesibilitas komunikasi bagi individu dengan disabilitas tunarungu dan tunawicara di lingkungan pelayanan kesehatan.
- Memfasilitasi interaksi yang lebih baik antara penyandang disabilitas dan tenaga kesehatan, mengurangi kesenjangan komunikasi.
- Memberdayakan individu disabilitas untuk berpartisipasi aktif dalam masyarakat dengan meningkatkan kemampuan komunikasi mereka.
- Mengurangi ketergantungan pada penerjemah bahasa isyarat yang tidak selalu tersedia di setiap fasilitas kesehatan.
- Mendorong inklusi sosial dengan mempromosikan pemahaman dan dukungan terhadap keberagaman komunikasi.
- Menawarkan solusi teknologi yang praktis dan mudah diakses, meningkatkan kualitas hidup individu disabilitas melalui aksesibilitas komunikasi yang lebih baik.

5. Spesifikasi Teknis Aplikasi Bi-Cara

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_mail import Mail, Message
from flask_restx import Resource, Api, Namespace, reqparse
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy import func
from sqlalchemy.exc import SQLAlchemyError
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 mediapipe.framework.formats import landmark_pb2
from joblib import load
from tensorflow.keras.models import load_model
import keras
import mediapipe as mp
import jwt
import base64
import cv2
import numpy as np
import tensorflow as tf
import random
import time
import os
import imghdr
import re
import warnings
import pytz
import logging
```

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 autentikasi, dan OpenCV untuk pemrosesan gambar. Pustaka lainnya seperti base64, numpy, random, dan pytz juga diimpor untuk berbagai utilitas tambahan.

2. Konfigurasi Database dan SMTP

```
#initialization_database_mysql
app.config["SQLALCHEMY_DATABASE_URI"] = "mysql://root:@127.0.0.1:3306/bicara_db"
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.config["SQLALCHEMY_ECHO"] = True
db = SQLAlchemy(app)

#Initialization_email_mailtrap
app.config['MAIL_SERVER'] = 'sandbox.smtp.mailtrap.io'
app.config['MAIL_PORT'] = 2525
app.config['MAIL_USERNAME'] = 'e3c1eafad1ec53'
app.config['MAIL_PASSWORD'] = '7c4bf20a63f6b2'
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USE_SSL'] = False
mail = Mail(app)
```

Gambar 2 Konfigurasi Database dan SMTP

Kode di atas menginisialisasi basis data MySQL dan konfigurasi email untuk aplikasi Flask. Konfigurasi basis data dilakukan dengan mengatur URI MySQL pada `SQLALCHEMY_DATABASE_URI`, menonaktifkan pelacakan modifikasi dengan `SQLALCHEMY_TRACK_MODIFICATIONS`, dan mengaktifkan logging SQL dengan `SQLALCHEMY_ECHO`. Setelah itu, objek `SQLAlchemy` diinisialisasi dengan aplikasi Flask. Untuk konfigurasi email, digunakan Mailtrap sebagai server email dengan mengatur `MAIL_SERVER`, `MAIL_PORT`, `MAIL_USERNAME`, dan `MAIL_PASSWORD`. Enkripsi TLS diaktifkan dengan `MAIL_USE_TLS`, sementara SSL dinonaktifkan dengan `MAIL_USE_SSL`. Objek `Mail` kemudian diinisialisasi dengan aplikasi Flask.

3. Route dan API - Daftar

```

parser4Register = reqparse.RequestParser()
parser4Register.add_argument('name', type=str, location='json', required=True, help='Masukkan Nama')
parser4Register.add_argument('gender_id', type=int, location='json', required=True, help='Masukkan Id Gender')
parser4Register.add_argument('email', type=str, location='json', required=True, help='Masukkan Email')
parser4Register.add_argument('password', type=str, location='json', required=True, help='Masukkan Password')

@api.route('/register', methods=['POST'])
class Register_Route(Resource):
    @api.expect(parser4Register, validate=True)
    @api.response(201, 'Created')
    def post(self):
        args = parser4Register.parse_args()
        email = args['email']
        now = datetime.now(local_timezone)
        print("waktu saat ini", now)
        try:
            with db.session.begin():
                userCount = User.query.filter_by(email=email).count()
                if userCount > 0:
                    return {'message': 'Email sudah digunakan'}, 400
                user = User()
                user.name = args['name']
                user.gender_id = args['gender_id']
                user.email = email
                user.password = generate_password_hash(args['password'])
                db.session.add(user)
                db.session.flush()
                otp = random.randint(1000, 9999)
                print("Int kode otp", otp)
                otp_expired_at = now + timedelta(minutes=5)
                mOtp = Otp()
                mOtp.user_id = user.id
                mOtp.otp = generate_password_hash(str(otp))
                mOtp.otp_expired_at = otp_expired_at
                db.session.add(mOtp)
                sendMail(email, str(otp))
        except Exception as e:
            return {'message': str(e)}, 500
        return {
            'message': 'Berhasil mengirim OTP ke email '+ user.email,
            'data': {
                'user_id': user.id,
                'otp_expired_at': mOtp.otp_expired_at.strftime('%Y-%m-%d %H:%M:%S')
            }
        }, 201

```

Gambar 3 Route dan API - Daftar

Kode di atas mendefinisikan endpoint API untuk registrasi pengguna baru dengan metode POST menggunakan kelas `Register_Route` dari Flask-RESTx. Saat endpoint `/register` diakses, fungsi `post` memvalidasi dan menguraikan argumen permintaan menggunakan `parser4Register`, lalu memeriksa apakah email sudah digunakan.

Jika email belum digunakan, fungsi membuat objek `User` baru, menghasilkan kode OTP acak 4 digit, menyimpannya di tabel `Otp` dengan waktu kedaluwarsa lima menit, dan mengirimkan OTP ke email pengguna. Jika terjadi kesalahan, perubahan di-rollback dan pesan kesalahan dikembalikan dengan status 500. Jika berhasil, pesan sukses dan data pengguna beserta waktu kedaluwarsa OTP dikembalikan dengan status 201.

4. Route dan API - Verifikasi Kode OTP

```
parser4VerifyOtp = reqparse.RequestParser()
parser4VerifyOtp.add_argument('api-key', type=str, location='headers', required=True, help='Masukkan API-KEY')
parser4VerifyOtp.add_argument('otp', type=str, location='json', required=True, help='Masukkan Kode OTP')
parser4VerifyOtp.add_argument('user_id', type=int, location='json', required=True, help='Masukkan User Id')

@api.route('/verify-otp')
class VerifyOtpRoute(Resource):
    @api.expect(parser4VerifyOtp, validate=True)
    @api.response(200, 'Verified')
    def post(self):
        args = parser4VerifyOtp.parse_args()
        apiKey = args['api-key']
        otp = args['otp']
        user_id = args['user_id']
        now = datetime.now(local_timezone)
        print('waktu saat ini', now)
        if API_KEY != apiKey:
            return {'message': 'API KEY Invalid!'}, 400
        try:
            with db.session.begin():
                checkOtp = otp.query.filter(otp.user_id==user_id).first()
                if not check_password_hash(checkOtp.otp, otp):
                    return {'message': 'OTP Invalid!'}, 400
                if checkOtp.otp_expired_at < datetime.now():
                    return {'message': 'OTP Expired!'}, 400
                user = User.query.get(checkOtp.user_id)
                if not user:
                    return {'message': 'User not found!'}, 400
                user.verify_otp = True
                user.updated_at = now
                user_data = {
                    'user_id': user.id,
                    'nama': user.name,
                    'email': user.email,
                    'verify_otp': user.verify_otp,
                    'created_at': user.created_at.strftime('%Y-%m-%d %H:%M:%S'),
                    'updated_at': user.updated_at.strftime('%Y-%m-%d %H:%M:%S'),
                }
                print(user_data)
        except Exception as e:
            return {'message': str(e)}, 500
        return {
            'message': 'Berhasil Verifikasi OTP',
            'data': user_data
        }, 201
```

Gambar 4 Route dan API - Verifikasi Kode OTP

Kode di atas mendefinisikan endpoint API untuk verifikasi OTP menggunakan metode POST pada rute '/verify-otp'. Sebuah 'RequestParser' bernama 'parser4VerifyOtp' digunakan untuk menguraikan dan memvalidasi argumen yang dikirim dalam permintaan, termasuk API key, OTP, dan user_id. Saat permintaan POST diterima, fungsi 'post' mengambil argumen ini dan mencatat waktu saat ini. Kode kemudian memverifikasi validitas API key. Jika API key tidak valid, respons dengan pesan kesalahan dan status 400 dikembalikan.

Jika API key valid, transaksi database dimulai. Kode memeriksa apakah OTP yang diberikan cocok dengan yang disimpan di database dan apakah OTP tersebut belum kedaluwarsa. Jika OTP tidak valid atau telah kedaluwarsa, pesan kesalahan dengan status 400 dikembalikan. Jika OTP valid, pengguna yang terkait dengan OTP diperoleh dari database. Status verifikasi OTP pengguna diperbarui dan data pengguna terbaru dikumpulkan. Jika terjadi kesalahan selama proses ini, perubahan akan di-rollback secara otomatis dan pesan kesalahan dikembalikan dengan status 500. Jika berhasil, fungsi mengembalikan pesan sukses beserta data pengguna dengan status 201.

5. Route dan API - Login

```
@api.route('/login')
class Login_Route(Resource):
    @api.expect(parser4Login, validate=True)
    @api.response(200, 'OK')
    def post(self):
        args = parser4Login.parse_args()
        email = args['email']
        password = args['password']
        apiKey = args['api-key']
        now = datetime.now(local_timezone)
        print("waktu saat ini", now)
        if API_KEY != apiKey:
            return {'message': 'API KEY Invalid!'}, 400
        try:
            with db.session.begin():
                user = User.query.filter_by(email=email).first()
                if not user:
                    return {'message': 'Kamu belum daftar, yuk daftar dulu!'}, 400
                if not check_password_hash(user.password, password):
                    return {'message': 'Password yang kamu masukkan salah!'}, 400
                exp = now + timedelta(hours=5)
                print("waktu exp: ", exp)
                payload = {
                    'req_body': str(user.id) + user.name + str(user.gender_id) + user.email + user.password,
                    'aud': AUDIENCE_MOBILE,
                    'iss': ISSUER,
                    'iat': int(time.time()),
                    'exp': exp
                }
                token = jwt.encode(payload, SECRET_KEY, algorithm='HS256')
                print("ini token jwt: ", token)
                print("panjang token", len(token))
                mAuth = Auth.query.filter_by(user_id=user.id).first()
                if not mAuth:
                    mAuth = Auth()
                    mAuth.user_id = user.id
                    mAuth.token = token
                    mAuth.token_expired_at = exp
                    mAuth.login_at = now
                    mAuth.created_at = now
                    db.session.add(mAuth)
                else:
                    mAuth.token = token
                    mAuth.token_expired_at = exp
                    mAuth.login_at = now
                    mAuth.updated_at = now
                user_data = {
                    'user_id': user.id,
                    'name': user.name,
                    'email': user.email,
                    'login_at': mAuth.login_at.strftime('%Y-%m-%d %H:%M:%S'),
                    'created_at': mAuth.created_at.strftime('%Y-%m-%d %H:%M:%S'),
                    'updated_at': mAuth.updated_at.strftime('%Y-%m-%d %H:%M:%S') if mAuth.updated_at else None
                }
                print(user_data)
            except Exception as e:
                return {'message': str(e)}, 500
            return {
                'message': 'Berhasil login',
                'token': mAuth.token,
                'data': user_data
            }, 201
```

Gambar 5 Route dan API - Login

Kode di atas mendefinisikan endpoint API untuk login pengguna menggunakan metode POST pada rute '/login' dengan kelas 'Login_Route' dari Flask-RESTx. Fungsi 'post' memvalidasi dan menguraikan argumen permintaan menggunakan 'parser4Login', yang mencakup API key, email, dan password. Setelah memverifikasi validitas API key, fungsi mencari pengguna berdasarkan email di database dan memverifikasi kata sandi. Jika email dan kata sandi valid, fungsi membuat token JWT dengan masa berlaku lima jam dan menyimpannya dalam tabel 'Auth' bersama dengan informasi login. Jika terjadi kesalahan, transaksi di-rollback dan pesan kesalahan dikembalikan dengan status 500. Jika berhasil, fungsi mengembalikan pesan sukses, token JWT, dan data pengguna dengan status 201.

6. Route dan API - GET Profil

```
parser4Profile = reqparse.RequestParser()
parser4Profile.add_argument('user_id', type=int, location='args', required=True, help='Masukkan User Id')

@api.route('/get-profile')
class GetProfile_Route(Resource):
    @api.expect(parser4Profile, validate=True)
    @api.doc(security='Bearer')
    @api.response(200, 'OK')
    def get(self):
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        args = parser4Profile.parse_args()
        user_id = args['user_id']

        is_valid = isTokenValid(jwtToken)
        print(is_valid)
        if is_valid:
            getUser = User.query.filter_by(id=user_id).first()
            gender_id = getUser.gender_id
            gender = Gender.query.filter_by(id=gender_id).first()

            return {
                'data': {
                    'user_id': getUser.id,
                    'name': getUser.name,
                    'email': getUser.email,
                    'gender': gender.gender
                }
            }, 200
        else:
            return {'message': 'Token tidak valid, silahkan masuk dulu'}, 401
```

Gambar 6 Route dan API - GET Profil

Kode di atas mendefinisikan endpoint API untuk mendapatkan profil pengguna menggunakan metode GET pada rute '/get-profile' dengan kelas 'GetProfile_Route' dari Flask-RESTx. Fungsi 'get' memvalidasi argumen permintaan 'user_id' menggunakan 'parser4Profile' dan memeriksa token JWT yang dikirim dalam header 'Authorization'.

Jika token valid, fungsi mengambil data pengguna dari database berdasarkan 'user_id' dan mengambil informasi jenis kelamin pengguna dari tabel 'Gender'. Data pengguna, termasuk ID, nama, email, dan jenis kelamin, dikembalikan dengan status 200. Jika token tidak valid, pesan kesalahan dikembalikan dengan status 401.

7. Route dan API - Ubah Profil

```
parser4EditProfile = reqparse.RequestParser()
parser4EditProfile.add_argument('user_id', type=int, location='json', required=True, help='Masukkan User Id')
parser4EditProfile.add_argument('name', type=str, location='json', required=True, help='Masukkan Nama')
parser4EditProfile.add_argument('gender', type=int, location='json', required=True, help='Masukkan Gender')

@api.route('/edit-profile')
class EditProfile Route(Resource):
    @api.expect(parser4EditProfile, validate=True)
    @api.doc(security='Bearer')
    @api.response(200, 'OK')
    def put(self):
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        args = parser4EditProfile.parse_args()
        user_id = args['user_id']
        name = args['name']
        gender = args['gender']

        now = datetime.now(local_timezone)
        print("waktu saat ini", now)

        is_valid = isTokenValid(jwtToken)
        if is_valid:
            getUser = User.query.filter_by(id=user_id).first()
            getUser.name = name
            getUser.gender = gender
            getUser.updated_at = now
            db.session.commit()
            return {
                'message': 'Berhasil update data user',
                'data': {
                    'name': getUser.name,
                    'gender': getUser.gender,
                    'updated_at': getUser.updated_at.strftime('%Y-%m-%d %H:%M:%S')
                }
            }, 200
        else:
            return {'message': 'Token tidak valid, silahkan masuk dulu'}, 401
```

Gambar 7 Route dan API - Ubah Profil

Kode di atas mendefinisikan endpoint API untuk mengedit profil pengguna menggunakan metode PUT pada rute `/edit-profile` dengan kelas `EditProfile_Route` dari Flask-RESTx. Fungsi `put` memvalidasi argumen permintaan `user_id`, `name`, dan `gender` menggunakan `parser4EditProfile`, dan memeriksa token JWT yang dikirim dalam header `Authorization`.

Jika token valid, fungsi mengambil data pengguna dari database berdasarkan `user_id`, memperbarui nama dan jenis kelamin pengguna, serta mencatat waktu pembaruan. Perubahan disimpan ke database dengan `db.session.commit()`. Jika berhasil, fungsi mengembalikan pesan sukses dan data pengguna terbaru dengan status 200. Jika token tidak valid, pesan kesalahan dikembalikan dengan status 401.

8. Route dan API - Ubah Password

```
parser4EditPassword = reqparse.RequestParser()
parser4EditPassword.add_argument('user_id', type=int, location='json', required=True, help='Masukkan User Id')
parser4EditPassword.add_argument('new_pass', type=str, location='json', required=True, help='Masukkan Password Baru')
parser4EditPassword.add_argument('confir_pass', type=str, location='json', required=True, help='Konfirmasi Password Baru')

@api.route('/edit-password')
class EditPasswordRoute(Resource):
    @api.expect(parser4EditPassword, validate=True)
    @api.doc(security='Bearer')
    @api.response(200, 'OK')
    def put(self):
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        args = parser4EditPassword.parse_args()
        user_id = args['user_id']
        new_pass = args['new_pass']
        confir_pass = args['confir_pass']
        now = datetime.now(local_timezone)
        print('waktu saat ini: ', now)
        is_valid = isTokenValid(jwtToken)
        if is_valid:
            if new_pass != confir_pass:
                return {'message': 'Password baru tidak sama'}, 400

            getUser = User.query.filter_by(id=user_id).first()
            getUser.password = generate_password_hash(new_pass)
            getUser.updated_at = now
            db.session.commit()
            return {
                'message': 'Berhasil ubah password',
                'user_id': getUser.id,
                'updated_at': getUser.updated_at.strftime('%Y-%m-%d %H:%M:%S')
            }, 200
        else:
            return {'message': 'Token tidak valid, silahkan masuk dulu'}, 401
```

Gambar 8 Route dan API - Ubah Password

Kode di atas mendefinisikan endpoint API untuk mengubah kata sandi pengguna menggunakan metode PUT pada rute '/edit-password' dengan kelas 'EditPasswordRoute' dari Flask-RESTx. Fungsi 'put' memvalidasi argumen permintaan 'user_id', 'new_pass', dan 'confirm_pass' menggunakan 'parser4EditPassword', serta memeriksa token JWT yang dikirim dalam header 'Authorization'.

Jika token valid, fungsi memeriksa apakah 'new_pass' dan 'confirm_pass' sama. Jika tidak, pesan kesalahan dengan status 400 dikembalikan. Jika sama, fungsi mengambil data pengguna dari database berdasarkan 'user_id', memperbarui kata sandi dengan yang baru di-hash, dan mencatat waktu pembaruan. Perubahan disimpan ke database dengan 'db.session.commit()'. Jika berhasil, fungsi mengembalikan pesan sukses, ID pengguna, dan waktu pembaruan dengan status 200. Jika token tidak valid, pesan kesalahan dikembalikan dengan status 401.

9. Route dan API - Feedback

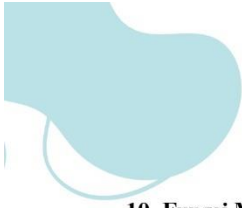
```
parser4Feedback = reqparse.RequestParser()
parser4Feedback.add_argument('user_id', type=int, location='json', required=True, help='Masukkan User Id')
parser4Feedback.add_argument('feedback', type=str, location='json', required=True, help='Masukkan Feedback')

@api.route('/feedback')
class Feedback_Route(Resource):
    @api.expect(parser4Feedback, validate=True)
    @api.doc(security='Bearer')
    @api.response(200, 'OK')
    def post(self):
        auth = request.headers.get('Authorization')
        jwtToken = auth[7:]
        args = parser4Feedback.parse_args()
        user_id = args['user_id']
        feedback = args['feedback']
        now = datetime.now(local_timezone)
        print("waktu saat ini", now)
        ts_valid = tsTokenValid(jwtToken)
        if ts_valid:
            tFeedback = Feedback()
            tFeedback.user_id = user_id
            tFeedback.feedback = feedback
            db.session.add(tFeedback)
            db.session.commit()
            return {
                'message': 'Berhasil memberikan feedback!',
                'created_at': tFeedback.created_at.strftime('%Y-%m-%d %H:%M:%S')
            }, 200
        else:
            return {'message': 'Token tidak valid, silahkan masuk dulu'}, 401
```

Gambar 9 Route dan API - Feedback

Kode di atas mendefinisikan endpoint API untuk mengirimkan umpan balik pengguna menggunakan metode POST pada rute '/feedback' dengan kelas 'Feedback_Route' dari Flask-RESTx. Fungsi 'post' memvalidasi argumen permintaan 'user_id' dan 'feedback' menggunakan 'parser4Feedback', serta memeriksa token JWT yang dikirim dalam header 'Authorization'.

Jika token valid, fungsi membuat objek 'Feedback', menyimpan ID pengguna dan umpan balik yang diterima, dan menyimpannya ke database dengan 'db.session.commit()'. Jika berhasil, fungsi mengembalikan pesan sukses dan waktu pembuatan umpan balik dengan status 200. Jika token tidak valid, pesan kesalahan dikembalikan dengan status 401.



10. Fungsi Mediapipe Hand Landmarker

Berikut ini merupakan source code fungsi yang digunakan untuk memberikan landmark yang terdapat tangan pada frame-frame flutter:

```
BaseOptions = mp.tasks.BaseOptions
HandLandmarker = mp.tasks.vision.HandLandmarker
HandLandmarkerOptions = mp.tasks.vision.HandLandmarkerOptions
VisionRunningMode = mp.tasks.vision.RunningMode

class LandmarkerAndResult:
    def __init__(self):
        self.result = mp.tasks.vision.HandLandmarkerResult
        self.landmarker = mp.tasks.vision.HandLandmarker
        self.createLandmarker()

    def createLandmarker(self):
        options = HandLandmarkerOptions(
            base_options=BaseOptions(model_asset_path='./model/hand_landmarker.task'),
            running_mode=VisionRunningMode.IMAGE,
            num_hands=2)

        self.landmarker = self.landmarker.create_from_options(options)

    def detect(self, frame):
        rgb_image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        mp_image = mp.Image(image_format=mp.ImageFormat.SRGB, data=rgb_image)
        self.result = self.landmarker.detect(image=mp_image)

    def close(self):
        self.landmarker.close()
```

Gambar 10 Fungsi Mediapipe Hand Landmarker

Kode di atas mengimpor dan menginisialisasi MediaPipe Hand Landmarker untuk deteksi tangan. Kelas `LandmarkerAndResult` bertanggung jawab untuk membuat dan mengelola objek hand landmarker serta mengambil hasil deteksi.

Dalam konstruktor, objek hand landmarker dibuat dengan opsi yang mencakup jalur model dan mode berjalan (dalam hal ini, mode gambar untuk mendeteksi hingga dua tangan). Metode `detect` mengonversi frame input ke format RGB, kemudian menggunakan landmarker untuk mendeteksi tangan dalam gambar. Metode `close` digunakan untuk menutup objek landmarker saat tidak lagi diperlukan.






11. Fungsi Ekstrak Koordinat Pada Frame

Berikut ini merupakan source code fungsi untuk mengekstrak koordinat yang telah di deteksi oleh mediapipe hand landmarker.

```
def extractCoordinates(resultDetect):
    coordinates = []
    hand_landmarks_list = resultDetect.hand_landmarks
    handedness_list = resultDetect.handedness
    for idx in range(len(hand_landmarks_list)):
        hand_landmarks = hand_landmarks_list[idx]
        handedness = handedness_list[idx]
        hand_label = handedness[0].category_name
        coordinates.append(hand_label)
        for i, landmark in enumerate(hand_landmarks):
            coordinates.append(landmark.x)
            coordinates.append(landmark.y)
            coordinates.append(landmark.z)
    return coordinates

def preprocess_landmarks(coordinates, scaler):
    for i, val in enumerate(coordinates):
        if val == 'Right':
            coordinates[i] = 1
        elif val == 'Left':
            coordinates[i] = 0
    coordinates = [float(val) for val in coordinates]
    newData = np.array([coordinates])
    newData_scaled = scaler.transform(newData)
    newDataReshape = newData.reshape(newData_scaled.shape[0], newData_scaled.shape[1], 1)
    return newDataReshape

def split_coordinates(coordinates):
    list1 = coordinates[:64]
    list2 = coordinates[64:]
    return list1, list2
```



Gambar 11 Fungsi Ekstrak Koordinat Pada Frame

Kode di atas berisi tiga fungsi yang berfokus pada pengolahan koordinat landmark hasil deteksi tangan menggunakan MediaPipe.

Fungsi `extractCoordinates(resultDetect)`:

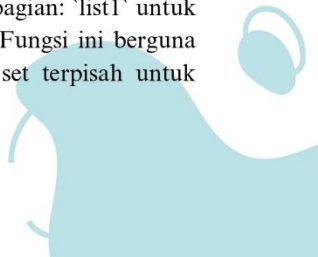
Fungsi ini mengekstrak koordinat landmark tangan dari hasil deteksi. Ia mengumpulkan daftar landmark dan informasi tangan (kiri/kanan), lalu menyimpan label tangan serta posisi x, y, dan z dari setiap landmark dalam daftar `coordinates`.

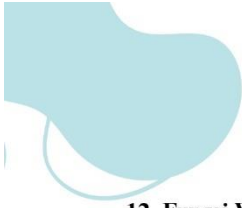
Fungsi `preprocess_landmarks(coordinates, scaler)`:

Fungsi ini memproses koordinat yang diekstrak sebelum digunakan untuk prediksi dengan model LSTM. Ia mengonversi label tangan menjadi nilai numerik (1 untuk kanan, 0 untuk kiri), mengubah semua nilai menjadi float, dan melakukan penskalaan menggunakan objek `scaler`. Hasil akhir direshape menjadi format yang sesuai untuk model.

Fungsi `split_coordinates(coordinates)`:

Fungsi ini membagi daftar koordinat menjadi dua bagian: `list1` untuk 64 elemen pertama dan `list2` untuk sisa elemen. Fungsi ini berguna untuk memisahkan data landmark menjadi dua set terpisah untuk analisis lebih lanjut.





12. Fungsi Visualisasi Hand Landmark Pada Frame

Berikut ini merupakan source code fungsi untuk frame yang sudah terdeteksi handlandmarkernya dan di beri visualisasi landmarknya.

```
def draw_landmarks_on_image(rgb_image, detection_result: mp.tasks.vision.HandLandmarkerResult):
    try:
        if detection_result.hand_landmarks == []:
            return rgb_image
        else:
            hand_landmarks_list = detection_result.hand_landmarks
            annotated_image = np.copy(rgb_image)

            for idx in range(len(hand_landmarks_list)):
                hand_landmarks = hand_landmarks_list[idx]

                hand_landmarks_proto = landmark_pb2.NormalizedLandmarkList()
                hand_landmarks_proto.landmark.extend([
                    landmark_pb2.NormalizedLandmark(x=landmark.x, y=landmark.y, z=landmark.z)
                    for landmark in hand_landmarks])
                mp.solutions.drawing_utils.draw_landmarks(
                    annotated_image,
                    hand_landmarks_proto,
                    mp.solutions.hands.HAND_CONNECTIONS,
                    mp.solutions.drawing_styles.get_default_hand_landmarks_style(),
                    mp.solutions.drawing_styles.get_default_hand_connections_style())
            return annotated_image
    except:
        return rgb_image
```

Gambar 12 Fungsi Visualisasi Hand Landmark Pada Frame

Fungsi `draw_landmarks_on_image` bertujuan untuk menggambar landmark tangan yang terdeteksi pada gambar RGB. Langkah-langkahnya adalah sebagai berikut:

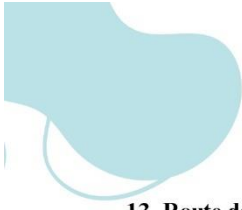
Memeriksa Landmark Tangan: Fungsi mengecek apakah `detection_result.hand_landmarks` kosong. Jika kosong, fungsi mengembalikan gambar asli tanpa modifikasi.

Membuat Salinan Gambar: Jika terdapat landmark, fungsi membuat salinan dari gambar RGB asli untuk anotasi.

Menggambar Landmark Tangan: Untuk setiap set landmark tangan yang terdeteksi, fungsi membuat objek `hand_landmarks_proto` dari tipe `NormalizedLandmarkList` dan menambahkan koordinat landmark tangan. Selanjutnya, menggunakan `mp.solutions.drawing_utils.draw_landmarks`, landmark tangan dan koneksinya digambar pada gambar yang disalin, dengan gaya default untuk tampilan yang konsisten.

Mengembalikan Gambar Anotasi: Setelah menggambar semua landmark, fungsi mengembalikan gambar yang telah dianotasi. Jika terjadi kesalahan, gambar asli dikembalikan tanpa perubahan.





13. Route dan API - Klasifikasi Frame dari Flutter

Berikut ini merupakan source code untuk setiap frame yang dikirimkan dari flutter ke webservice untuk diklasifikasikan.

```
hand_landmarker = LandmarkerAndResult()
@api.route('/detect_landmarks', methods=['POST'])
class Detect_Landmarks_Route(Resource):
    def post(self):
        try:
            if 'image' not in request.files:
                return {"error": "Tidak ada file gambar yang diunggah"}, 400
            image_file = request.files['image']
            imgdata = cv2.imdecode(np.frombuffer(image_file.read(), np.uint8), cv2.IMREAD_COLOR)
            height, width, _ = imgdata.shape
            print(f"Ukuran gambar: {width}x{height}")
            if height > width:
                print("Gambar dalam orientasi potret (portrait).")
            else:
                print("Gambar dalam orientasi lanskap (landscape).")
            hand_landmarker.detect(imgdata)
            resultDetect = hand_landmarker.result
            print("hasil landmark: ", resultDetect)

            if resultDetect is not None and hasattr(resultDetect, 'hand_landmarks'):
                print("halo")
                coordinates = extractCoordinates(resultDetect)
                print("Koordinat: ", coordinates)
                preLandmarks = None
                frameImg = imgdata
                frame = None
                if len(coordinates) > 0 and len(coordinates) <= 64:
                    preLandmarks = preprocess_landmarks(coordinates, scaler)
                    frame = draw_landmarks_on_image(frameImg, resultDetect)
                elif len(coordinates) >= 64:
                    list1, list2 = split_coordinates(coordinates)
                    preLandmarks = preprocess_landmarks(list1, scaler)
                    preLandmarks = preprocess_landmarks(list2, scaler)
                    frame = draw_landmarks_on_image(frameImg, resultDetect)
                    frame = draw_landmarks_on_image(frameImg, resultDetect)
                else:
                    return {"noLabel": "Tidak ada landmark tangan yang terdeteksi"}, 202
                frame = np.array(frame, dtype=np.uint8)
                if preLandmarks is not None:
                    prediction = model.predict(preLandmarks)
                    print("Prediksi: ", prediction)
                    predicted_class = np.argmax(prediction, axis=1)
                    print("Kelas yang diprediksi: ", predicted_class)
                    percentage_probabilities = prediction[0][predicted_class] * 100
```

Gambar 13 Preprocessing Frame dan Prediksi

Kode ini mendefinisikan endpoint /detect_landmarks yang menerima permintaan POST untuk memproses gambar yang diunggah dan mendeteksi landmark tangan. Gambar yang diunggah dikonversi dari buffer ke format gambar menggunakan OpenCV, dan dimensinya diperiksa untuk menentukan orientasi. Objek hand_landmarker digunakan untuk mendeteksi landmark tangan pada gambar, dan hasil deteksi disimpan dalam resultDetect. Jika hasil deteksi berisi landmark tangan, koordinat landmark diekstrak dan dipreproses menggunakan fungsi extractCoordinates dan preprocess_landmarks. Berdasarkan jumlah koordinat yang ditemukan, gambar dengan landmark yang digambar dihasilkan. Jika tidak ada landmark yang terdeteksi, pesan kesalahan dikembalikan.



```
label = ...
if predicted_class == 0:
    label = "an"
elif predicted_class == 1:
    label = "atur"
elif predicted_class == 2:
    label = "bagaimana"
elif predicted_class == 3:
    label = "daftar"
elif predicted_class == 4:
    label = "dengan"
elif predicted_class == 5:
    label = "dokter"
elif predicted_class == 6:
    label = "jadwal"
elif predicted_class == 7:
    label = "kan"
elif predicted_class == 8:
    label = "kapan"
elif predicted_class == 9:
    label = "me"
elif predicted_class == 10:
    label = "minum"
elif predicted_class == 11:
    label = "obat"
elif predicted_class == 12:
    label = "proses"
elif predicted_class == 13:
    label = "saya"
else:
    label = "Tidak dikenal"
    print(label)
    _, buffer = cv2.imencode('.jpg', frame)
    landmarked_image = base64.b64encode(buffer).decode('utf-8')
    return {
        'label': label,
        'probability': percentage_probabilities.tolist(),
        'landmarked_image': landmarked_image
    }, 201
else:
    return {"error": "Tidak ada landmark tangan yang diproses"}, 400
else:
    return {"error": "Tidak ada landmark tangan yang terdeteksi"}, 400
except Exception as e:
    return {"error": str(e)}, 400
```

Gambar 14 Kirim Response Fame dari Hasil Klasifikasi

Setelah data landmark dipreproses, model LSTM digunakan untuk memprediksi kelas gerakan tangan. Hasil prediksi dikonversi menjadi label dan probabilitas yang sesuai. Gambar dengan landmark yang digambar dikonversi kembali ke format JPEG dan diubah menjadi string base64 untuk dikembalikan dalam respons API. Jika semua langkah berhasil, respons API akan berisi label, probabilitas, dan gambar dengan landmark, dikembalikan dengan status 201. Jika terjadi kesalahan selama proses, pesan kesalahan dikembalikan dengan status 400.



14. Deteksi Secara Realtime di Flutter

```
void _startStreaming() {
  if (_controller != null && !_controller!.value.isInitialized) {
    if (!_isStreaming) {
      setState(() {
        _isStreaming = true;
      });
      _controller!.startImageStream(CameraImage image) {
        _frameCount++;
        if (_frameCount % 60 == 0) {
          print('Frame count: $_frameCount');
          final convertedImage = convertYUV420ToImage(image);
          final jpegBytes = img.encodeJpg(convertedImage);
          _sendImageToServer(jpegBytes);
        }
      } else {
        setState(() {
          _isStreaming = false;
        });
        _controller!.stopImageStream();
      }
    }
  }
}

img.Image convertYUV420ToImage(CameraImage cameraImage) {
  final int width = cameraImage.width;
  print('lebar: $width');
  final int height = cameraImage.height;
  print('tinggi: $height');
  final yBuffer = cameraImage.planes[0].bytes;
  final uBuffer = cameraImage.planes[1].bytes;
  final vBuffer = cameraImage.planes[2].bytes;
  final int yRowStride = cameraImage.planes[0].bytesPerRow;
  final int yPixelStride = cameraImage.planes[0].bytesPerPixel;
  final int uvRowStride = cameraImage.planes[1].bytesPerRow;
  final int uvPixelStride = cameraImage.planes[1].bytesPerPixel;
  final image = img.Image(width: width, height: height);
  for (int h = 0; h < height; h++) {
    int uvh = (h / 2).floor();
    for (int w = 0; w < width; w++) {
      int row = (w / 2).floor();
      final yIndex = h * yRowStride + w * yPixelStride;
      final uvIndex = uvh * uvRowStride + row * uvPixelStride;
      final y = yBuffer[yIndex];
      final u = uBuffer[uvIndex];
      final v = vBuffer[uvIndex];
      int r = (y + v * 1436 / 1824 - 179).round().clamp(0, 255);
      int g = (y - u * 46549 / 131072 + 46 - v * 93604 / 131072 + 9)
        .round()
        .clamp(0, 255);
      int b = (y + u * 1814 / 1824 - 227).round().clamp(0, 255);
      image.setPixel(w, h, img.ColorRgb8(r, g, b));
    }
  }
  if (height < width) {
    print('Gambar dalam orientasi lanskap, memutar gambar.');
```

Gambar 15 Fungsi Start Realtime dan Capture Frame

Fungsi `_startStreaming()` menginisiasi atau menghentikan streaming gambar dari kamera. Jika streaming belum dimulai, fungsi ini akan mengaktifkannya dengan memanggil `_controller!.startImageStream()`. Setiap frame ke-60, gambar akan diubah menjadi format JPEG menggunakan fungsi `convertYUV420ToImage()`, kemudian dikirim ke server dengan `_sendImageToServer()`. Jika streaming sudah aktif, fungsi ini akan menghentikan streaming dengan `_controller!.stopImageStream()`.

Fungsi `convertYUV420ToImage()` mengonversi gambar dari format YUV420 ke gambar RGB. Fungsi ini mengambil data dari tiga buffer (Y, U, V) dan menggabungkannya untuk membuat gambar RGB. Gambar yang dihasilkan akan diputar 90 derajat jika lebih lebar daripada tinggi, untuk mengubah orientasi lanskap ke potret.



```

Future<void> _sendImageToServer(List<int> imageBytes) async {
  final uri = Uri.parse('${config.base_url}/detect_landmarks');
  final request = http.MultipartRequest('POST', uri)
    ..files.add(http.MultipartFile.fromBytes(
      'image',
      imageBytes,
      filename: 'image.jpg',
      contentType: MediaType('image', 'jpeg'),
    ));

  final response = await request.send();
  if (response.statusCode == 201) {
    final responseData = await response.stream.bytesToString();
    final jsonResponse = jsonDecode(responseData);
    setState(() {
      _label = jsonResponse['label'];
      _landmarkedImageBytes = base64Decode(
        jsonResponse['landmarked_image']); // Decode base64 image
    });
    _showLandmarkedImage();
    print('Predicted Label: $_label');
    print('Landmark: $_landmarkedImageBytes');
    String audioPath = getAudioPath(_label);
    await _audioPlayer.play(AssetSource(audioPath));
  } else if (response.statusCode == 202) {
    final responseData = await response.stream.bytesToString();
    final jsonResponse = jsonDecode(responseData);
    setState(() {
      _label = jsonResponse['noLabel'];
    });
    print('Predicted Label: $_label');
    await _audioPlayer
      .play(AssetSource('audio/ttsmaker-file-2024-7-7-17-44-28.mp3'));
  } else {
    print('Error: ${response.statusCode}');
  }
}

```

Gambar 16 Integrasi API Deteksi Klasifikasi SIBI

Fungsi `_sendImageToServer` mengirim gambar dalam format JPEG ke server untuk mendeteksi landmark tangan. Gambar dikirim melalui permintaan HTTP POST ke URL yang ditentukan. Jika respons dari server berhasil (status kode 201), data yang diterima didekodekan dari format JSON, termasuk label prediksi dan gambar dengan landmark yang ditandai. Label dan gambar ditampilkan di aplikasi, dan file audio yang sesuai dengan label dimainkan. Jika respons dari server adalah status kode 202, yang menunjukkan tidak ada landmark terdeteksi, pesan yang sesuai ditampilkan dan file audio default dimainkan. Jika terjadi kesalahan, status kode ditampilkan di konsol.

Fungsi `getAudioPath` menentukan jalur file audio berdasarkan label prediksi yang diterima dari server. Fungsi ini menggunakan struktur `switch-case` untuk mencocokkan label prediksi dengan file audio yang sesuai. Jika label tidak ada dalam daftar, fungsi mencetak pesan di konsol dan mengembalikan string kosong. Fungsi ini memungkinkan pemutaran audio yang spesifik berdasarkan hasil prediksi dari server, mendukung fitur feedback audio di aplikasi.

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202468368, 19 Juli 2024

Pencipta

Nama : **Desty Nurul Anitsa, Dyah Apriliani dkk**

Alamat : Desa Pegirikan RT 32 RW 08, Kecamatan Talang, Kabupaten 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 Mobile Bi-Cara Menggunakan Algoritma
Long Short Term Memory Dalam Berkomunikasi Dengan Bahasa
Isyarat**

Tanggal dan tempat diumumkan untuk pertama kali
di wilayah Indonesia atau di luar wilayah Indonesia : 19 Juli 2024, di Tegal

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

Nomor pencatatan : 000643718

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



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b

Direktur Hak Cipta dan Desain Industri

IGNATIUS M.T. SILALAH
NIP. 196812301996031001

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

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Desty Nurul Anitsa	Desa Pegirikan RT 32 RW 08, Kecamatan Talang, Kabupaten Tegal Provinsi Jawa Tengah 52193, Talang, Tegal
2	Dyah Apriliani	Perumahan Mutiara Indah Block C4, Jl. Nangka Gg. 2 RT 02 RW 02, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal Provinsi Jawa Tengah 52412, Slawi, Tegal
3	Hepatika Zidny Ilmadina	Jalan Kenanga Gang 1 Nomor 9, Kelurahan Mangkukusuman, Kecamatan Tegal Timur Kota Tegal Provinsi Jawa Tengah 52123, Tegal Timur, Tegal





SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN SKRIPSI

Nama : Desty Nurul Anltsa
Nim : 20090134
No. Ponsel : 0895422825549
Judul TA : Pengembangan applikasi *Mobile* Bi-Cara menggunakan algoritma *Long Short Term Memory* dalam berkomunikasi dengan bahasa isyarat
Dosen Pembimbing I : Dyah Aprilliani, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	25 Maret 2024	Aplikasi	Memperbanyak dataset video SIBI	
2.	17 April 2024	Aplikasi	Membuat model SIBI	
3.	20 Mei 2024	Aplikasi	Membuat aplikasi mobile	
4.	02 Juni 2024	Aplikasi	Integrasi Backend Webservice dengan Front End	
5.	04 Juli 2024	Aplikasi	Integrasi Model dengan Aplikasi Bi-Cara	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
6.	03 Juli 2024	Aplikasi	Perbaikan Model Aplikasi	
7.	09 Juli 2024	Aplikasi, HKI	Perbaikan Model Aplikasi	
8.	16 Juli 2024	HKI	Manual book dan Dokumen Teknis	
9.	18 Juli 2024	Laporan	Laporan Akhir	
10.	22 Juli 2024	Laporan	Revisi Laporan Akhir	
11.	24 Juli 2024	Laporan	Revisi Laporan Akhir	
11.	25 Juli 2024	Laporan	ACC Laporan Akhir	

Tegal, 25 Juli 2024
Dosen Pembimbing I

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



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**





LEMBAR BIMBINGAN SKRIPSI

Nama : Desty Nurul Anitsa
Nim : 20090134
No. Ponsel : 0895422825549
Judul TA : Pengembangan aplikasi *Mobile Bi-Cara* menggunakan algoritma *Long Short Term Memory* dalam berkomunikasi dengan bahasa isyarat
Dosen Pembimbing I : Hepatika Zidny Ilmadina, S.Pd., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	25 Maret 2024	Aplikasi	Memperbanyak dataset video SIBI	
2.	17 April 2024	Aplikasi	Membuat model SIBI	
3.	20 Mei 2024	Aplikasi	Membuat aplikasi mobile	
4.	02 Juni 2024	Aplikasi	Integrasi Backend Webservice dengan Front End	
5.	0 Juli 2024		Integrasi Model dengan Aplikasi Bi-Cara	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
5.	03 Juli 2024	Aplikasi	Perbaikan Model Aplikasi	
6.	09 Juli 2024	Aplikasi	Perbaikan Model Aplikasi	
7.	16 Juli 2024	Aplikasi, HKI	Manual book dan Dokumen Teknis	
8.	18 Juli 2024	Laporan	Laporan Akhir	

Tegal, 18 Juli 2024
Dosen Pembimbing II



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