

IMPLEMENTASI *COMPUTER VISION* PADA APLIKASI MEDIA PENGENALAN BAHASA ISYARAT BERBASIS *ANDROID*

Muhammad Risman¹, Samirah Rahayu², Sri Farida Utami³

Program Studi Teknik Komputer, Politeknik Sukabumi
Jl. Babakan Sirna No.25, Benteng, Kec. Warudoyong, Kota Sukabumi, Jawa Barat 43132
samirah@polteksmi.ac.id

Abstrak

Perkembangan penggunaan *smartphone* sangat pesat saat ini, khususnya pada *smartphone* dengan sistem operasi *android*. Aplikasi ini dirancang untuk membantu pengguna, untuk mengenal dan mempelajari bahasa isyarat, dalam mengenali dan mempelajari gerakan tangan yang mewakili huruf abjad. Aplikasi ini mampu mendeteksi dan mengklasifikasikan gerakan tangan secara real-time, memberikan *feedback* berupa teks dan suara untuk memfasilitasi pengenalan dan pembelajaran. Implementasi *Computer Vision* pada aplikasi media pengenalan bahasa isyarat berbasis *Android* bertujuan untuk mempermudah proses pengenalan dan pembelajaran bagi pengguna yang membutuhkan. Dalam penelitian ini, penulis menggunakan 6.500 data citra, masing-masing terdiri dari 250 gambar untuk setiap abjad, guna melatih model pengenalan bahasa isyarat. Hasil dari data latih menunjukkan tingkat akurasi yang sangat tinggi, yaitu mencapai 97%. Jarak ideal untuk pengenalan adalah 30 cm dengan rata-rata akurasi 0.97. Pada intensitas cahaya terang sebesar 454 *Lux*, rata-rata akurasi deteksi mencapai 0.98, menunjukkan performa terbaik dari sistem di bawah pencahayaan optimal. Hal ini mengindikasikan bahwa sistem mampu mendeteksi dengan sangat akurat ketika diberikan pencahayaan yang memadai.

Kata kunci: *Computer Vision*, Bahasa Isyarat, Android, Pengenalan Citra, SIBI

I. PENDAHULUAN

Perkembangan teknologi informasi yang pesat mencakup perangkat keras, perangkat lunak, serta teknologi komunikasi yang memungkinkan pengiriman informasi dengan cepat, seperti telepon pintar Android. Perkembangan ini memberikan dampak signifikan dalam dunia pendidikan, memungkinkan transfer ilmu pengetahuan tanpa batasan jarak dan waktu, serta memfasilitasi proses pembelajaran melalui aplikasi berbasis Android [1]. Salah satu aplikasi yang dikembangkan adalah pembelajaran bahasa isyarat menggunakan metode BISINDO (Bahasa Isyarat Indonesia), yang dikembangkan oleh dan untuk masyarakat tuna rungu [2]. BISINDO menjadi pilihan utama karena lebih efektif dan mudah digunakan. Sistem Isyarat Bahasa

Indonesia (SIBI), yang mengadopsi dari American Sign Language (ASL) [3]. Aplikasi ini bertujuan untuk menjembatani kesenjangan komunikasi antara penyandang tunarungu, tunawicara, dan masyarakat umum, dengan harapan dapat mempermudah pemahaman di antara kedua pihak [4].

Penelitian yang dilakukan sebelumnya oleh Arif Iqbal dan Rika Rosnelly Teknologi ilmu pengetahuan yang berkembang saat ini sangat berpengaruh pada kehidupan manusia. Hal ini menjadikan teknologi sebagai kebutuhan dalam mempermudah aktivitas manusia sehari-hari, seperti aktivitas pembelajaran, pemerintahan, perkantoran, hiburan dan lainnya. Salah satu diantaranya adalah aktivitas pembelajaran, dengan adanya diterapkannya teknologi yang canggih tentunya dapat membantu aktivitas dalam sistem pembelajaran [5].

Merancang aplikasi pengenalan bahasa isyarat Indonesia berbasis Android untuk tujuan pengenalan. Pengembangan yang dilakukan mencakup proses menangkap gambar dari bentuk tangan pengguna, melakukan preprocessing data, menjalankan inferensi model untuk mengenali gestur, dan menampilkan hasil dalam bentuk teks dan suara. Penelitian ini berusaha mengisi gap dalam teknologi pembelajaran bahasa isyarat Indonesia dengan menawarkan pendekatan interaktif yang lebih efektif untuk memperkenalkan bahasa isyarat kepada pengguna.

II. METODE PENELITIAN

Metode penelitian diawali dengan studi literatur kemudian dilanjutkan dengan tahap analisis kebutuhan sistem, tahap perancangan sistem, tahap implementasi sistem, dan terakhir pengujian aplikasi. Gambaran alur penelitian ditunjukkan pada gambar 1 dibawah ini.



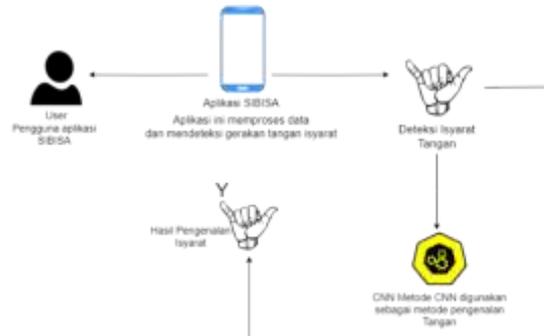
Gambar 1. Alur Penelitian

Pada gambar 1, proses analisis kebutuhan diperlukan untuk mengamati apa saja kebutuhan dari pengguna. Analisis tersebut didasarkan atas latar belakang penelitian sehingga berbagai kebutuhan fungsional dan kebutuhan non fungsional dapat terpenuhi. Perancangan sistem meliputi proses pelatihan model machine learning, pembuatan data citra, dan desain antarmuka aplikasi. Untuk implementasi model akan dimasukkan kedalam sistem mobile android dan setelah itu tahap terakhir adalah pengujian aplikasi kepada pengguna.

A. Arsitektur Sistem Aplikasi

1. Arsitektur Sistem

Pengguna harus membuka aplikasi penerjemah bahasa isyarat terlebih dahulu. Kemudian, pengguna harus mengaktifkan kamera pada aplikasi tersebut. Setelah kamera aktif, pengguna diarahkan untuk mengarahkan kamera ke teman atau orang yang sedang melakukan peragaan bahasa isyarat abjad. Mengelola proses tangkapan, *preprocessing*, inferensi model, dan menampilkan hasil terjemahan berupa teks dan suara. Gambaran arsitektur sistem ditunjukkan pada gambar 2 dibawah ini.



Gambar 2. Arsitektur Sistem

2. Hand gesture recognition

Hand gesture recognition atau yang disebut pula pengenalan isyarat tangan merupakan interaksi antara manusia dan komputer (mesin). Hal yang paling mendasar dalam pengenalan isyarat tangan adalah dengan membuat interaksi yang alami (sesuai kebiasaan) antara manusia dan komputer yang mana pengenalan isyarat tersebut dapat digunakan untuk mengontrol sebuah robot atau menyampaikan suatu informasi. Sistem arsitektur mengenai hand gesture ini adalah penangkapan gambar, pergerakan tangan (*hand tracking*) dan pengenalan isyarat tangan dan *control mouse*. Sistem pengenalan isyarat tangan (*hand gesture recognition*) dapat digunakan untuk beberapa pengaplikasian seperti terjemahan, *virtual environment*, *smart surveillance*, kontrol robot, sistem medis dan lain – lain [6].

3. Library Mediapipe

Mediapipe adalah kerangka kerja yang memungkinkan pengembangan untuk membangun saluran ML multi-modal (video, audio, seri, waktu apa pun). Sebagai kerangka node dan tepi atau landmark, mereka melacak titik titik kunci diberbagai bagian tubuh. Semua titik koordinat dinormalisasi tiga dimensi [7].

Dengan menggunakan *multithreading* dan GPU *acceleration*, membuat *MediaPipe* sangat cepat dalam memproses banyak data yang masuk, *MediaPipe* mempunyai kalkulator yang akan berjalan ketika program di mulai dan akan berhenti ketika program selesai [5].

Kalkulator tersebut di gunakan *MediaPipe* untuk konsep *multithreading*, *MediaPipe* menggunakan *machine learning* secara *real-time*. *Machine learning* yang ditawarkan oleh *MediaPipe* antara lain *Face Detection*, *Face Mesh*, *Iris*, *Hands*, *Pose*, *Holistic*, *Hair Segmentation*, *Object Detection*, *Box Tracking*, *Instant Motion Tracking* [8].

4. Google Colab

Google Colab adalah sebuah *IDE* untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh server *Google* yang memiliki perangkat keras dengan performa yang tinggi. Dari sisi perangkat lunak, *Google Colab* telah menyediakan hampir sebagian besar pustaka (*library*) yang dibutuhkan. Dalam penelitian ini pustaka yang dibutuhkan adalah Keras, *TensorFlow*, *NumPy*, *Pandas*, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat *Matplotlib*. Bahkan seluruh versi disediakan, misalnya *TensorFlow* versi 1.x maupun 2.x tersedia, begitu juga versi *Python* yang mulai dari versi 2.x hingga 3.x.

Dari sisi perangkat keras, *Google Colab* menyediakan layanan berupa media penyimpanan yang terintegrasi dengan *Google Drive*, prosesor yang berupa CPU, GPU, dan TPU, serta RAM. Dengan jaminan kemampuan servernya yang stabil hampir keseluruhan pemrosesan tidak menemukan kendala dengan *Google Colab* selama koneksi jaringan internet lancar [9].

Colab memungkinkan siapa saja menulis dan mengeksekusi kode *python* arbitrer melalui *browser*, dan sangat cocok untuk *machine learning*, analisis data, serta pendidikan. Secara lebih teknis, *Colab* merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk GPU. *Resource Colab* tidak dijamin dan sifatnya terbatas, serta batas penggunaannya terkadang berfluktuasi. Hal ini diperlukan agar *Colab* dapat menyediakan *resource* secara gratis. Pengguna yang ingin memiliki akses lebih andal ke *resource* yang lebih baik dapat menggunakan *Colab Pro*. Memperkenalkan *Colab Pro* merupakan Langkah pertama yang *Google* ambil untuk melayani pengguna yang ingin melakukan lebih banyak hal di *Colab*.

Tujuan jangka panjang pihak *Google* adalah untuk terus menyediakan versi gratis *Colab*, dan di saat yang bersamaan berkembang secara berkelanjutan untuk memenuhi kebutuhan pengguna *Google* [10].

5. Computer Vision

Computer vision adalah sebuah kemampuan sebuah *computer* yang di desain agar mampu melihat sebuah *object* sehingga mampu menampilkan onbjek digital dan bisa mengoleksi data secara *visual* komputer bisa melakukan beberapa pekerjaan yang tidak bisa dilakukan oleh manusia :

- *Computer* mampu melihat data dalam bentuk *pixel* bahkan dalam warna yang berbeda.
- *Computer* mampu membandingkan dua *object* gambar yang sama persis. *Computer* mampu melihat sebuah *object* data selama berjam-jam bahkan berhari – hari. *Vision* itu sendiri adalah suatu proses evaluasi sebuah data yang bersumber dari *image* umumnya *camera*, dengan teknik ekstrasi menggunakan algoritma tertentu [11].

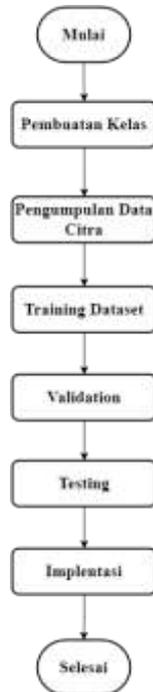
Artificial Intelligence atau kecerdasan buatan adalah suatu simulasi dari kecerdasan manusia yang dimodelkan didalam mesin dan diprogram agar mampu berfikir layaknya manusia. Implementasi dari teknologi AI telah banyak digunakan, salah satu cabang ilmu pengetahuan dari *Artificial Intelligence* yaitu *computer vision* yang mempelajari disiplin ilmu tentang bagaimana komputer dapat mengenali objek yang diamati [12]. Tujuannya adalah untuk membuat mesin yang dapat "melihat" dunia seperti manusia dan mampu mengambil keputusan berdasarkan informasi visual tersebut. Salah satu penerapan dari bidang *computer vision* ini dengan melakukan pengawasan misalnya untuk melakukan pendeteksian objek kendaraan [13].

6. Keras

Keras merupakan perangkat lunak jaringan yang berbasis *open source*, dan ditulis dengan menggunakan bahasa pemrograman *Python*. Keras juga dapat dijalankan menggunakan *MXNet*, *Tensorflow*, *Deeplearning4j*, dan *Theano* yang khusus dirancang guna mempercepat eksperimen yang berhubungan dengan *Deep Learning*. Fitur yang menonjol dari Keras adalah *Tensorflow* dan *Theano* digunakan sebagai *backend*. Keras juga dapat berjalan lancar di

kedua CPU dan GPU. Keras mendukung hampir semua model jaringan saraf mulai dari *convolutional*, *pooling*, *recurrent*, *embedding*, dan lainnya. Selanjutnya, model ini dapat dikombinasikan untuk membangun model yang lebih kompleks [14].

B. Proses Training Dataset



Gambar 3. Alur Pelatihan Citra

Pada gambar 4 merupakan alur proses dari pelatihan data citra. Proses tersebut dilakukan secara bertahap dari awal sampai akhir. Berikut merupakan penjelasan lebih detail dari gambar 4.

1. Pembuatan Kelas

Kelas tersebut mencakup gestur tangan yang mewakili huruf A hingga Z. Setiap kelas ini akan berisi data citra yang sesuai dengan gestur masing-masing huruf alfabet, yang nantinya digunakan untuk melatih model agar mampu mengenali setiap huruf dengan benar.

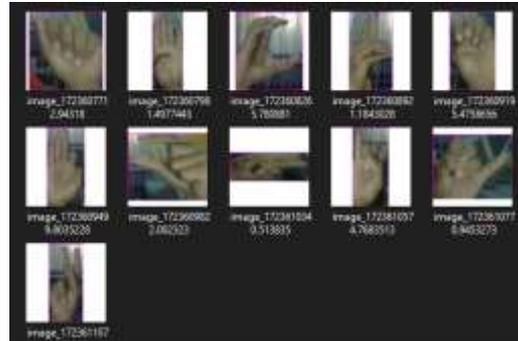


Gambar 4. Pembuatan Kelas

2. Pengumpulan Data Citra

Proses pengumpulan data yang telah dilakukan secara mandiri oleh penulis. Dalam pengumpulan data ini, penulis sendiri yang melakukan capture gambar untuk memastikan

bahwa dataset yang dihasilkan sesuai dengan kebutuhan. Setiap data dikumpulkan dengan teliti, sehingga dapat digunakan untuk lebih lanjut dalam penelitian ini. Pengumpulan dataset ini merupakan langkah penting untuk mendapatkan data yang akurat. Contoh kumpulan data citra untuk dilatih pada gambar 5 dibawah.



Gambar 5. Data Citra SIBI

3. Training Data Cita

Tahap pelatihan model ini dilakukan dengan membagi dua folder yaitu test dan train proses pembagian ini telah digenerate secara otomatis. Pada bagian ini dilakukan dengan menggunakan Google Colab yaitu salah satu workspace pelatihan model yang populer dan ringan digunakan.

```
data = gesture_recognizer.Dataset.from_folder(
    dirname=dataset_path,
    fparams=gesture_recognizer.HandDataPreprocessingParams()
)

train_data, rest_data = data.split(0.8)

validation_data, test_data = rest_data.split(0.5)

Downloading https://storage.googleapis.com/mediavision-datasets/gesture_recognizer/172360771
Downloading https://storage.googleapis.com/mediavision-datasets/gesture_recognizer/172360772
/usr/local/lib/python3.10/dist-packages/google/protobuf/symbol_database.py:55: UserWarning: warnings.warn("`symbol_database.getprototype()` is deprecated. Please "
```

Gambar 6. Proses Training Dataset

Pada gambar 6 merupakan proses berlangsungnya pelatihan datasets yang dilakukan dengan bantuan GPU dari google dan bawaan dari laptop. Setelah selesai melakukan pelatihan maka akan dihasilkan model machine learning berekstensi .task.

Proses selanjutnya yaitu validate model dan testing model sudah otomatis dilakukan pada proses training berlangsung dan menghasilkan tingkat akurasi gambar 5 dibawah.

```
loss, acc = model.evaluate(test_data, batch_size=16)
print("[INFO] accuracy: {:.2f}%".format(acc * 100))
print("[INFO] Loss: {}".format(loss))

5/5 [=====] - 3s 33ms/step
[INFO] accuracy: 97.01%
[INFO] Loss: 0.08422502875328064
```

Gambar 7. Akurasi Training

C. Perancangan Aplikasi Android

1. Gambaran Umum Aplikasi

Penelitian ini mengembangkan aplikasi penerjemah Sistem Isyarat Bahasa Indonesia (SIBI) berbasis Android. Tujuan dari aplikasi ini yaitu sebagai tujuan pengenalan. Selain itu juga memiliki fitur sebagai media pembelajaran bahasa isyarat sibi.

2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan proses yang harus dilakukan oleh sistem berinteraksi dengan pengguna. Kebutuhan fungsional dapat dilihat pada Tabel 1.

Tabel 1. Kebutuhan Fungsional Aplikasi.

No	Kebutuhan Fungsional
1	Aplikasi harus memiliki akses ke kamera perangkat untuk mengambil gambar dari tangan bahasa isyarat.
2	Mengidentifikasi objek yang ada di dalam video atau gambar menggunakan teknik deteksi objek.
3	Fitur yang memperkenalkan alfabet (A-Z) dan angka (0-10) dalam bahasa isyarat.
4	Daftar kata-kata umum dan frasa dalam bahasa isyarat beserta demonstrasi visual.
5	Gambar atau animasi yang menunjukkan langkah-langkah gerakan isyarat.

Pada Tabel 1 menampilkan kebutuhan fungsional aplikasi. Dalam prosesnya memiliki tiga fitur utama yaitu fitur terjemah bahasa isyarat, fitur animasi yang memperkenalkan alfabet (A-Z) dan angka (0-10) dalam bahasa isyarat., dan Daftar kata-kata umum dan frasa dalam bahasa isyarat beserta demonstrasi visual.

3. Kebutuhan Non Fungsional

Kebutuhan Non-Fungsional Kebutuhan non-fungsional adalah kebutuhan yang memerlukan properti untuk mendukung sistem sehingga dapat berjalan dengan baik. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Kebutuhan non fungsionalitas terbagi menjadi beberapa analisis, yaitu perangkat keras dan

perangkat lunak. .Kebutuhan non fungsional dapat dilihat pada Tabel 2.

Tabel 2. Kebutuhan Non Fungsional Aplikasi

No	Hardware	Spesifikasi	Software
1	Laptop Lenovo Thinkpad T440s	Processor : Intel Core i5-4300U Ram : 12Gb SSD : 512Gb	Visual Studio Code. Android Studio
2	Android Redmi Note 8	Processor : Octa-core Max2.01GHz. Ram : 4Gb. Kamera : 48MP	

III. HASIL DAN PEMBAHASAN

A. Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan aplikasi desain interface yaitu menggunakan Android Studio dengan bahasa pemrograman java.

Aplikasi ini memiliki 7 antarmuka yang terdiri dari halaman splash screen, halaman utama, halaman terjemah, halaman huruf dan angka, halaman kata, halaman profil pengembang, dan halaman tentang aplikasi. Berikut pada gambar 10 dan 11 contoh gambaran dari antarmuka aplikasi.



Gambar 8. Implementasi Antarmuka Aplikasi 1



Gambar 9. Implementasi Antarmuka Aplikasi 2

Pada gambar 10 dan 11 menampilkan berbagai antarmuka aplikasi. Implementasi antarmuka berdasarkan kebutuhan fungsional dan non fungsional.

B. Pengujian Antarmuka Pengguna

Pengujian antarmuka terdiri dari pengujian fitur aplikasi yang dilakukan dengan metode

blackbox. Berikut merupakan Tabel 3 hasil pengujian antarmuka pada aplikasi.

Tabel 3. Hasil Pengujian Antarmuka

Aktifitas Pengujian	Hasil Pengujian	Keterangan
Membuka aplikasi	Menampilkan halaman splash screen dan halaman utama	Berhasil
Membuka menu terjemah	Menampilkan halaman terjemah bahasa isyarat	Berhasil
Membuka menu huruf dan angka	Menampilkan halaman huruf dan angka	Berhasil
Membuka menu kata	Menampilkan halaman kata isyarat	Berhasil
Membuka menu profil pengembang	Menampilkan halaman profil pengembang	Berhasil
Membuka menu tentang aplikasi	Menampilkan menu tentang aplikasi	Berhasil

C. Pengujian Akurasi Pengenalan Abjad

Pengujian deteksi model bahasa isyarat dilakukan secara manual dengan menggunakan aplikasi secara langsung. Proses pengujian dilakukan dengan cara mengarahkan kamera dari ponsel aplikasi ke arah peraga isyarat sibi. Berikut pada gambar 10 merupakan contoh hasil deteksi yang berhasil.



Gambar 10. Contoh Hasil Deteksi Isyarat

Pada gambar 12 merupakan tampilan dari halaman terjemah isyarat. Proses pengambilan deteksi harus melakukan access permission ke kamera pengguna. Setelah diizinkan maka sistem akan langsung mendeteksi isyarat. Ketika diarahkan ke tangan peraga.

Pada Tabel 4.2 menampilkan hasil deteksi huruf isyarat menggunakan aplikasi penerjemah bahasa isyarat yang sedang diuji. Setiap baris dalam tabel ini mencatat data mengenai huruf

isyarat yang terdeteksi, akurasi deteksi. Tingkat akurasi diukur dalam skala 0 hingga 1, di mana nilai 1.00 menunjukkan bahwa aplikasi mendeteksi huruf isyarat dengan akurasi 100%.

Tabel 4. Hasil Pengujian Akurasi

Abjad	Nilai Akurasi
A	1.00
B	1.00
C	0.98
D	0.92
E	0.91
F	0.99
G	0.98
H	1.00
I	1.00
J	1.00
K	1.00
L	1.00
M	0.80
N	1.00
O	0.89
P	0.76
Q	0.89
R	0.97
S	0.96
T	0.88
U	0.92
V	1.00
W	1.00
X	0.97
Y	1.00
Z	0.96

Kesimpulan dari tabel 4 hasil pengujian akurasi data huruf abjad bahasa isyarat, sistem menunjukkan performa yang sangat baik dengan akurasi sempurna (1.00) untuk huruf A, B, H, I, J, K, L, N, V, W, dan Y, menandakan deteksi yang sangat konsisten dan akurat. Huruf-huruf C, F, G, R, X, dan Z juga mencapai akurasi tinggi (0.96 - 0.99), meskipun ada beberapa kesalahan minor yang mungkin disebabkan oleh kemiripan gerakan dengan huruf lain. Sebaliknya, huruf D, E, O, T, U, dan Q memiliki akurasi yang masih tinggi tetapi menunjukkan kesalahan yang lebih sering (0.85 - 0.92), kemungkinan karena kompleksitas bentuk tangan. Huruf M dan P memiliki akurasi terendah (0.76 - 0.80), menunjukkan kesulitan signifikan dalam mendeteksi gerakan tangan untuk huruf-huruf ini, mungkin akibat bentuk yang lebih kompleks.

IV. KESIMPULAN

Penulis telah berhasil merancang aplikasi pengenalan bahasa isyarat berbasis Android yang berjalan cukup baik berdasarkan beberapa hasil pengujian. Aplikasi ini menunjukkan akurasi yang memadai dengan menggunakan 6.500 data citra, di mana setiap huruf abjad diwakili oleh 250 gambar. Dari pengujian jarak, jarak ideal adalah 30 cm dengan nilai rata-rata akurasi 0,97. Namun, pada intensitas cahaya rendah sebesar 3,6 Lux, beberapa huruf abjad tidak terdeteksi. Sebagai saran, disarankan untuk menambahkan fitur latihan yang memungkinkan pengguna mempraktikkan isyarat dengan umpan balik atau tes, menyediakan opsi berbagi animasi atau kata isyarat melalui media sosial, serta mendukung bahasa isyarat dari berbagai negara dengan opsi beralih antar sistem bahasa isyarat.

REFERENSI

- [1] S. Arfida, H. Wibowo, and A. F. Setya, "Penerapan Teknologi Android Terhadap Aplikasi Panduan Penggunaan Software Adobe Audition," *IJCCS*, vol. x, No.x, pp. 1–5.
- [2] I. Kautsar, R. Indra Borman, A. Sulistyawati, T. H. Informatika STMIK TEKNOKRAT Bandar Lampung Jl Zainal Abidin Pagaralam No, and L. Ratu Bandar Lampung, "APLIKASI PEMBELAJARAN BAHASA ISYARAT BAGI PENYANDANG TUNA RUNGU BERBASIS ANDROID DENGAN METODE BISINDO," pp. 6–8, 2015.
- [3] M. Huruf *et al.*, "Perancangan Media Pembelajaran Interaktif Bahasa Isyarat," vol. 15, no. 2, pp. 242–249, 2022, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom/page/242>
- [4] Z. Situmorang, A. Kamus, B. Isyarat...1, and H. Gunawan, "PERANCANGAN APLIKASI KAMUS BAHASA ISYARAT MENGGUNAKAN ALGORITMA LEVENSHTAIN SEBAGAI PENCARIAN KATA BERBASIS ANDROID DESIGNING A SIGN LANGUAGE DICTIONARY APPLICATION USING THE LEVENSHTAIN ALGORITHM AS AN ANDROID BASED WORD SEARCH."
- [5] M. A. , & R. R. Iqbal, "PERANCANGAN APLIKASI MEDIA PEMBELAJARAN PENGENALAN LAPISAN BUMI MENGGUNAKAN AUGMENTED REALITY BERBASIS ANDROID," 2020.
- [6] H. Yunita and E. Setyati, "Hand Gesture Recognition Sebagai Pengganti Mouse Komputer Menggunakan Kamera," *Jurnal ELTIKOM*, vol. 3, no. 2, pp. 64–76, Oct. 2019, doi: 10.31961/eltikom.v3i2.114.
- [7] U. A. Pringsewu, A. Fauziah Novianti, and Y. Saragih, "Aisyah Journal of Informatics and Electrical Engineering SISTEM IDENTIFIKASI PENGUKURAN BAJU MENGGUNAKAN HUMAN BODY ESTIMATION DATASET MEDIAPIPE DENGAN METODE EUCLIDEAN DISTANCE", [Online]. Available: <http://jti.aisyahuniversity.ac.id/index.php/AJIEE>
- [8] A. Hasyim Nur'azizan, A. Riqza Ardiansyah, and R. Fernandis, "Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe."
- [9] R. Gelar Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.
- [10] G. I. E. Soen, M. Marlina, and R. Renny, "Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants," *JITU: Journal Informatic Technology And Communication*, vol. 6, no. 1, pp. 24–30, Jun. 2022, doi: 10.36596/jitu.v6i1.781.
- [11] A. Purno and W. Wibowo, "Implementasi Teknik Computer Vision Dengan Metode Colored Markers Trajectory Secara Real Time," 2016.
- [12] J. Teknika, Y. T. Wulandari, and T. Rosandy, "Teknika 18 (2): 343-354 Implementasi Computer Vision Dalam

Sistem Deteksi Gerakan Disiplin
Kampus,” *IJCCS*, vol. x, No.x, pp. 1–5.

- [13] D. Indra, H. Herman, and F. S. Budi,
“Implementasi Sistem Penghitung
Kendaraan Otomatis Berbasis Computer
Vision,” *Komputika: Jurnal Sistem
Komputer*, vol. 12, no. 1, pp. 53–62, May
2023, doi:
10.34010/komputika.v12i1.9082.
- [14] R. Haris Alfikri *et al.*,
“PEMBANGUNAN APLIKASI
PENERJEMAH BAHASA ISYARAT
DENGAN METODE CNN BERBASIS
ANDROID,” 2022. [Online]. Available:
<https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>