

Sistem Deteksi Simbol Isyarat Pada SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan *Mediapipe* Berbasis *Android*

Rendra Soekarta¹, Suhardi Aras², Rezki^{3*}, Nofryanti Ainun K.D.P⁴

^{1, 2, 3, 4}Prodi Teknik Informatika, Universitas Muhammadiyah, Sorong

E-mail: rendrasoekarta@um-sorong.ac.id¹, suhardi.aras@gmail.com²,

rezki@um-sorong.ac.id^{3*}, nofryanti.ainun19@gmail.com⁴

Abstrak

Penyandang disabilitas, khususnya tunarungu, menghadapi tantangan besar dalam berkomunikasi, terutama di perguruan tinggi yang seringkali belum sepenuhnya inklusif. Penelitian ini bertujuan untuk mengembangkan sistem deteksi Bahasa Isyarat Indonesia (SIBI) berbasis Android menggunakan framework MediaPipe dan teknologi deep learning. Dataset penelitian mencakup 24 kelas huruf abjad statis yang diambil menggunakan kamera smartphone dan sumber daring. Proses pengembangan melibatkan ekstraksi fitur menggunakan MediaPipe dan integrasi ke platform Android. Pengujian menunjukkan model mencapai akurasi 90,85%, meskipun terdapat tantangan pada kelas dengan pola isyarat yang mirip. Hasil usability testing dengan 60 pengguna mencatat rata-rata kepuasan 83,51%, menunjukkan respons positif terhadap kemudahan penggunaan dan manfaat aplikasi. Aplikasi ini memungkinkan deteksi isyarat secara akurat dan cepat, serta stabil di berbagai kondisi. Hasil penelitian ini penting untuk meningkatkan aksesibilitas komunikasi penyandang tunarungu melalui solusi teknologi yang mudah diakses dan aplikatif.

Kata kunci— *Mediapipe, Deep Learning, Bahasa Isyarat, Android*

1. PENDAHULUAN

Pendidikan merupakan hak bagi setiap masyarakat Indonesia tidak terkecuali termasuk masyarakat dengan kebutuhan khusus atau penyandang disabilitas[1]. Di Indonesia, menurut Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional Pasal 5 Ayat 1, Setiap warga negara mempunyai hak yang sama untuk memperoleh pendidikan yang bermutu. Ini termasuk juga penyandang disabilitas yang harus diberikan kesempatan yang sama dalam memperoleh pendidikan di perguruan tinggi. Untuk mendukung hal ini, terdapat ketentuan dalam Pasal 42 ayat 3 yang menyatakan bahwa perguruan tinggi harus menyediakan lingkungan yang inklusif, menyediakan sarana dan sistem layanan yang memadai bagi penyandang disabilitas agar mereka dapat belajar dengan optimal.

Namun, dalam kenyataannya, masih banyak perguruan tinggi yang belum sepenuhnya siap menerima mahasiswa disabilitas [2], terutama dalam hal aksesibilitas komunikasi [3]. Penyandang disabilitas, khususnya tunarungu, menghadapi tantangan besar dalam berkomunikasi [4]. Tunarungu adalah gangguan pada pendengaran sehingga tidak dapat mendengar bunyi dengan sempurna atau bahkan tidak dapat mendengar sama sekali [5]. Bahasa isyarat adalah sarana utama bagi penyandang tunarungu untuk berinteraksi [6]. Namun, tidak semua orang mengerti atau dapat menggunakan bahasa isyarat, dan tidak semua lingkungan menyediakan aksesibilitas bagi orang dengan disabilitas pendengaran [7]. Hal ini menyebabkan hambatan komunikasi, Hambatan komunikasi ini berdampak negatif pada partisipasi sosial penyandang disabilitas dalam berbagai aspek kehidupan, termasuk Pendidikan [8].

Dengan demikian, diperlukan lebih banyak solusi teknologi dan strategi komunikasi yang dapat membantu memediasi interaksi antara penyandang tunarungu dan masyarakat umum yang

mayoritas menggunakan bahasa lisan. Sebelumnya telah dilakukan pengembangan sistem untuk klasifikasi abjad Bahasa Isyarat Indonesia (SIBI) menggunakan *Mediapipe* dan *metode Deep Learning*. Tujuan utama dari penelitian ini adalah untuk mempermudah komunikasi bagi penyandang tunarungu dan tunawicara dengan masyarakat umum. Penelitian ini menggunakan bantuan library *OpenCV* sebagai input gambar dari *webcam* dan library *Mediapipe* untuk ekstraksi fitur dari input *webcam*. Hasil akhir menunjukkan akurasi sebesar 94,32%.

Penelitian sebelumnya telah menunjukkan kemajuan signifikan dalam bidang deteksi bahasa isyarat menggunakan berbagai teknologi seperti pengolahan citra dan pengenalan pola dengan pemanfaatan model *Mediapipe*. Namun, berdasarkan penelitian sebelumnya solusi yang ada masih terbatas pada lingkungan desktop dan memerlukan perangkat keras yang canggih. Android adalah platform *open-source* yang sangat fleksibel, memungkinkan modifikasi dan distribusi oleh para pengembang. Dengan memanfaatkan platform Android, sistem deteksi bahasa isyarat ini dapat diakses oleh lebih banyak orang dan dapat diintegrasikan ke dalam aplikasi mobile yang telah menjadi bagian penting dari kehidupan sehari-hari masyarakat modern.

Berdasarkan permasalahan diatas maka akan diangkat judul tentang “DETEKSI SISTEM ISYARAT BAHASA INDONESIA (SIBI) MENGGUNAKAN MEDIAPIPE BERBASIS ANDROID” yang dapat mengidentifikasi makna dari isyarat yang diberikan melalui gerak oleh penyandang tunarungu berbasis android.

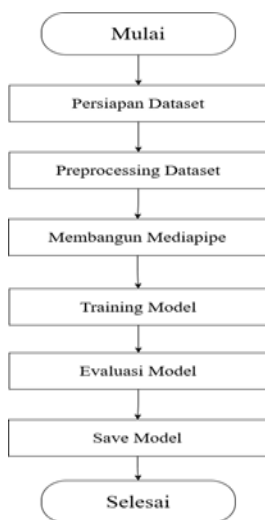
2. METODE PENELITIAN

2.1 Pengumpulan Dataset

Dataset yang digunakan pada berupa citra gambar manusia yang memeragakan isyarat simbol Abjad SIBI (Sistem Isyarat Bahasa Indonesia). Dataset berjumlah 24 kelas yaitu huruf A-Y. kelas yang dibuat tidak termasuk huruf J dan Z, karena kedua huruf tersebut menggunakan isyarat bergerak atau dinamis, sedangkan dataset gambar hanya memuat isyarat statis. Dataset merupakan gambar yang diambil sendiri menggunakan kamera smartphone, dan juga bersumber dari internet diambil dari *website kagle*

2.2 Pengembangan Mediapipe

MediaPipe adalah sebuah kerangka kerja sumber terbuka yang dikembangkan oleh *Google Inc.* dengan tujuan untuk mempermudah pengembang dalam membuat aplikasi multimodal yang mencakup video, gambar, audio, dan jenis data lainnya. Fokus utama dari framework ini adalah pada bidang computer vision, seperti kecerdasan buatan dan pemrosesan gambar [9] Alur pengembangan dapat dilihat pada gambar berikut.

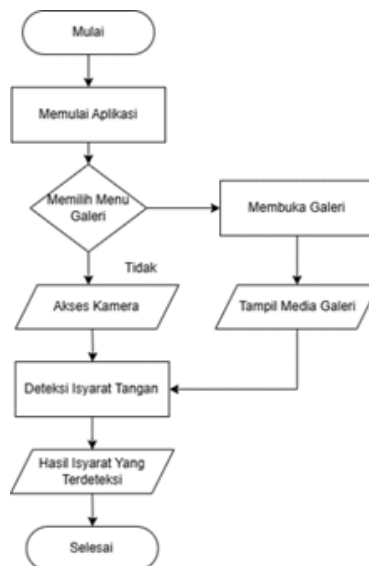


Gambar 1 Alur Pengembangan Model

Tahapan pengembangan model menggunakan *MediaPipe* untuk mendeteksi bahasa isyarat dimulai dengan persiapan dataset, yaitu mengunggah 24 kelas dataset huruf dalam bentuk gambar ke *Google Drive* untuk diakses melalui *Google Colab*. Selanjutnya, pada tahap preprocessing, gambar-gambar diubah ukurannya secara seragam, dan dilakukan data *cleaning* untuk memastikan kualitas dan keseimbangan dataset. Proses pembuatan model dimulai dengan instalasi *library* dan konfigurasi parameter yang sesuai. Setelah itu, dilakukan training model menggunakan data latih dan uji untuk mengukur performa. Evaluasi model dilakukan untuk menilai kemampuan mengenali bahasa isyarat, diikuti dengan perbaikan jika diperlukan. Terakhir, model yang telah diuji disimpan dalam format yang memungkinkan implementasi ke aplikasi Android.

2.3 Flowchart Sistem

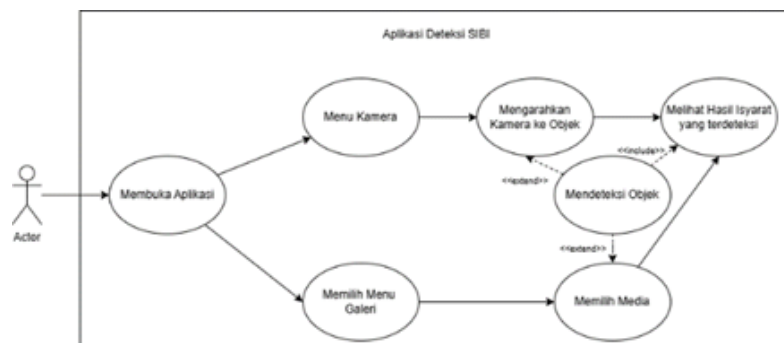
Flowchart adalah gambaran visual dari algoritma, di mana langkah-langkah komputasi direpresentasikan oleh simbol-simbol yang terhubung melalui panah yang menunjukkan aliran algoritma [10]. Representasi visual dari alur kerja aplikasi dapat dilihat pada gambar 3.



Gambar 2 Flowchart Sistem

2.4 Usecase

Use case adalah suatu representasi visual yang digunakan untuk menggambarkan perilaku sistem yang akan dirancang dengan memperlihatkan interaksi antara aktor-aktor yang terlibat dalam penggunaan sistem [11].



Gambar 3 Usecase Diagram

3. HASIL DAN PEMBAHASAN

3.1 Preparation dan Preprocessing Data

Pada tahap preparation, dataset yang disimpan pada google drive diakses dan dipanggil ke *google colab*. Selanjutnya *Preprocessing* data dimulai dengan transformasi data yang bertujuan untuk memeriksa dan memproses setiap gambar dalam folder dataset. Fungsi ini menyusuri semua folder dan subfolder di dalam dataset dan memastikan bahwa file yang diproses adalah gambar dengan ekstensi .png, .jpg, atau .jpeg. Setiap gambar kemudian dibaca dan dikonversi dari format warna BGR menjadi RGB. Selain itu, gambar diubah ukurannya menjadi 192x192 piksel untuk mencapai ukuran yang seragam. Terakhir Data cleaning adalah bagian untuk verifikasi struktur dataset dan menghitung jumlah gambar dalam setiap kelas. Fungsi ini memastikan bahwa setiap folder dalam path dataset adalah direktori yang mewakili kelas berbeda, lalu menghitung jumlah gambar dalam setiap kelas berdasarkan ekstensi file gambar. Verifikasi ini berguna untuk memastikan bahwa dataset memiliki struktur yang benar dan setiap kelas memiliki jumlah gambar yang mencukupi dan seimbang.

3.2 Membangun Model

Pada tahap pembangunan model, tahap awal dilakukan instalasi *library*. Selanjutnya membagi dataset menjadi data latih dan data uji. Data yang telah dibagi disimpan pada *directory* baru. Hasil pembagian dataset menjadi data latih dan data uji dapat dilihat pada gambar berikut.

```

Kelas A - Train: 278, Test: 70
Kelas B - Train: 268, Test: 68
Kelas C - Train: 229, Test: 58
Kelas D - Train: 212, Test: 54
Kelas E - Train: 269, Test: 68
Kelas F - Train: 284, Test: 71
Kelas G - Train: 276, Test: 69
Kelas H - Train: 264, Test: 66
Kelas I - Train: 268, Test: 68
Kelas K - Train: 271, Test: 68
Kelas L - Train: 256, Test: 64
Kelas M - Train: 260, Test: 65
Kelas N - Train: 266, Test: 67
Kelas O - Train: 240, Test: 61
Kelas P - Train: 208, Test: 53
Kelas Q - Train: 192, Test: 49
Kelas R - Train: 272, Test: 69
Kelas S - Train: 260, Test: 65
Kelas T - Train: 240, Test: 61
Kelas U - Train: 252, Test: 63
Kelas V - Train: 264, Test: 67
Kelas W - Train: 260, Test: 65
Kelas X - Train: 300, Test: 76
Kelas Y - Train: 256, Test: 64
Kelas none - Train: 0, Test: 0

```

Gambar 4 Hasil Pembagian Data

Selanjutnya mengatur parameter dengan nilai confidence minimal sebesar 0.5, membagi data latih menjadi 80% data latih dan 20% data validasi, lalu implementasi parameter pada masing-masing data, seperti pada gambar berikut.

```

from mediapipe_model_maker.python.vision import gesture_recognizer

handparams = gesture_recognizer.HandDataPreprocessingParams(
    min_detection_confidence=0.5
)
data = gesture_recognizer.Dataset.from_folder(str(dataset_train), handparams)
train_data, validation_data = data.split(0.8)

dataset_test = data_root / "test"
test_data = gesture_recognizer.Dataset.from_folder(
    str(dataset_test), handparams
)

```

Gambar 5 Mengatur Parameter

3.3 Training Model

Langkah selanjutnya adalah membuat dan melatih model *gesture recognizer*. Parameter HParams digunakan untuk menentukan konfigurasi model, seperti direktori ekspor model yang

mengatur lokasi penyimpanan model yang sudah dilatih, pada bagian ini juga telah ditentukan parameter lainnya. Parameter disini antara lain epochs = 30, dan batch size = 32, learning_rate = 0.001 dan lr_decay = 0.95. Proses training terlihat pada gambar berikut.

```
Epoch 15/30
131/131 [=====] - 7s 51ms/step - loss: 0.2273 - categorical_accuracy: 0.8907 - val_loss: 0.2707 - val_categorical_accuracy: 0.8876 - lr: 4.8767e-04
Epoch 16/30
131/131 [=====] - 5s 36ms/step - loss: 0.2268 - categorical_accuracy: 0.8872 - val_loss: 0.2670 - val_categorical_accuracy: 0.8914 - lr: 4.6329e-04
Epoch 17/30
131/131 [=====] - 7s 55ms/step - loss: 0.2211 - categorical_accuracy: 0.8860 - val_loss: 0.2656 - val_categorical_accuracy: 0.8886 - lr: 4.4913e-04
Epoch 18/30
131/131 [=====] - 5s 36ms/step - loss: 0.2165 - categorical_accuracy: 0.8929 - val_loss: 0.2641 - val_categorical_accuracy: 0.8895 - lr: 4.1812e-04
Epoch 19/30
131/131 [=====] - 7s 53ms/step - loss: 0.2122 - categorical_accuracy: 0.8938 - val_loss: 0.2639 - val_categorical_accuracy: 0.8867 - lr: 3.9721e-04
Epoch 20/30
131/131 [=====] - 5s 37ms/step - loss: 0.2088 - categorical_accuracy: 0.8989 - val_loss: 0.2620 - val_categorical_accuracy: 0.8905 - lr: 3.7735e-04
Epoch 21/30
131/131 [=====] - 14s 108ms/step - loss: 0.2091 - categorical_accuracy: 0.8943 - val_loss: 0.2603 - val_categorical_accuracy: 0.8886 - lr: 3.5849e-04
Epoch 22/30
131/131 [=====] - 6s 45ms/step - loss: 0.2106 - categorical_accuracy: 0.8943 - val_loss: 0.2591 - val_categorical_accuracy: 0.8905 - lr: 3.4056e-04
Epoch 23/30
131/131 [=====] - 5s 36ms/step - loss: 0.2066 - categorical_accuracy: 0.8936 - val_loss: 0.2590 - val_categorical_accuracy: 0.8905 - lr: 3.2352e-04
Epoch 24/30
131/131 [=====] - 8s 58ms/step - loss: 0.2037 - categorical_accuracy: 0.8960 - val_loss: 0.2577 - val_categorical_accuracy: 0.8914 - lr: 3.0736e-04
Epoch 25/30
131/131 [=====] - 5s 36ms/step - loss: 0.1999 - categorical_accuracy: 0.9027 - val_loss: 0.2565 - val_categorical_accuracy: 0.8914 - lr: 2.9199e-04
Epoch 26/30
131/131 [=====] - 5s 40ms/step - loss: 0.1995 - categorical_accuracy: 0.8974 - val_loss: 0.2558 - val_categorical_accuracy: 0.8914 - lr: 2.7739e-04
Epoch 27/30
131/131 [=====] - 5s 37ms/step - loss: 0.2036 - categorical_accuracy: 0.8974 - val_loss: 0.2549 - val_categorical_accuracy: 0.8933 - lr: 2.6352e-04
Epoch 28/30
131/131 [=====] - 8s 58ms/step - loss: 0.2038 - categorical_accuracy: 0.8960 - val_loss: 0.2530 - val_categorical_accuracy: 0.8886 - lr: 2.5034e-04
Epoch 29/30
131/131 [=====] - 5s 37ms/step - loss: 0.1978 - categorical_accuracy: 0.8953 - val_loss: 0.2533 - val_categorical_accuracy: 0.8924 - lr: 2.3783e-04
Epoch 30/30
131/131 [=====] - 5s 38ms/step - loss: 0.1987 - categorical_accuracy: 0.9055 - val_loss: 0.2526 - val_categorical_accuracy: 0.8924 - lr: 2.2594e-04
```

Gambar 6 Proses Training

3.4 Evaluasi Model

3.4.1 Akurasi

Hasil pelatihan model menunjukkan bahwa tingkat akurasi yang dicapai pada data latih adalah sebesar 90.85% yang dapat dilihat pada Gambar 7. Akurasi ini mengindikasikan bahwa model telah mempelajari pola-pola yang terkandung dalam data latih dengan cukup baik.

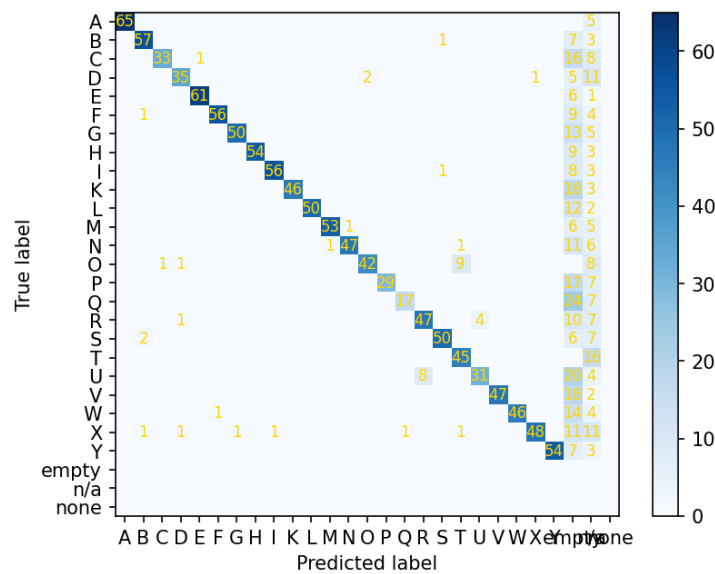
```
loss, acc = model.evaluate(test_data, batch_size=1)
print(f"Test loss: {loss:.4f}, Test accuracy: {acc:.2%}")

1301/1301 [=====] - 3s 2ms/step - loss: 0.2259 - categorical_accuracy: 0.9085
Test loss: 0.2259, Test accuracy: 90.85%
```

Gambar 7 Nilai Akurasi

3.4.2 Confusion Matriks

Pada tahap evaluasi model dilakukan pendekatan confusion matrik, hasil pada gambar berikut



Gambar 7 Confusion Matriks

Diagram menunjukkan model memiliki TP (True Point) yang tertinggi pada kelas “A” dengan jumlah 65 data menunjukkan bahwa sebagian besar data pada kelas ini terklasifikasi dengan baik. Sebaliknya, kelas “Q” dengan nilai FN (False Point) tertinggi dengan 24 data dengan titik temu empty menunjukkan data pada kelas ini banyak yang tidak dapat terdeteksi. Setelah menganalisis confusion matrix, kita dapat melanjutkan evaluasi model dengan menggunakan metrik precision, recall, dan F1-score untuk memahami performa klasifikasi model secara lebih detail terlihat pada gambar berikut.

Classification Report:

	precision	recall	f1-score	support
A	1.00	0.93	0.96	70
B	0.93	0.84	0.88	68
C	0.97	0.57	0.72	58
D	0.92	0.65	0.76	54
E	0.98	0.90	0.94	68
F	0.98	0.80	0.88	70
G	0.98	0.74	0.84	68
H	1.00	0.82	0.90	66
I	0.98	0.82	0.90	68
K	1.00	0.69	0.81	67
L	1.00	0.78	0.88	64
M	0.98	0.82	0.89	65
N	0.98	0.71	0.82	66
O	0.95	0.69	0.80	61
P	1.00	0.55	0.71	53
Q	0.94	0.35	0.52	48
R	0.95	0.68	0.76	69
S	0.96	0.77	0.85	65
T	0.80	0.74	0.77	61
U	0.89	0.49	0.63	63
V	1.00	0.70	0.82	67
W	1.00	0.71	0.83	65
X	0.98	0.63	0.77	76
Y	1.00	0.84	0.92	64
micro avg	0.96	0.72	0.83	1544
macro avg	0.96	0.72	0.82	1544
weighted avg	0.96	0.72	0.82	1544

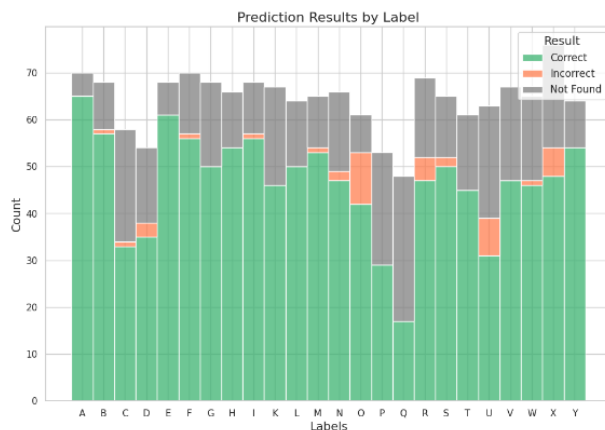
Average Metrics:
Precision (average): 0.8885
Recall (average): 0.6618
F1 Score (average): 0.7524

Gambar 8 Precision, Recall, F1-Score

Secara keseluruhan, model menunjukkan hasil yang cukup baik pada level rata-rata dengan precision rata-rata 0.88, yang mengindikasikan bahwa secara umum model jarang membuat kesalahan dalam klasifikasi positif. Namun, recall rata-rata sebesar 0.66 menunjukkan bahwa model masih kesulitan dalam mendeteksi seluruh data yang ada di setiap kelas, dengan beberapa kelas mengalami tingkat false negative yang lebih tinggi. Meskipun begitu, F1-score rata-rata 0.75 menunjukkan keseimbangan yang cukup baik antara precision dan recall di seluruh kelas.

3.4.3 Prediksi Label

Prediksi perkelas menampilkan gambaran banyaknya data yang terdeteksi dengan benar, salah mendeteksi dan tidak terdeteksi terlihat seperti pada gambar berikut.



Gambar 9 Prediksi Label

Sebagian besar klasifikasi yang salah berasal dari kelas yang memiliki pola yang mirip satu sama lain huruf O dan T, R dan U, dan seterusnya dapat dilihat pada gambar 13. Pada gambar terlihat pada kelas U sebanyak 30 data terdeteksi sebagai R.

label	pred	
O	T	9
U	R	8
R	U	4
D	O	2
S	B	2

Gambar 10 Incorrect Prediction

3.5 Save Model

Model yang telah dibuat dan dilatih kemudian diexport kedalam format task. Format khusus dalam mediapipe model maker untuk implementasi model ke android. Setelah itu dapat dilakukan visualisasi untuk melihat model mengidentifikasi dan membuat landmark. Hasil visualisasi dapat dilihat pada gambar berikut.



Gambar 11 Visualisai Model

3.6 Implementasi Aplikasi

Aplikasi yang dibuat menggunakan Bahasa pemrograman kotlin untuk implementasi pada android studio. Berikut merupakan hasil implemementasi model ke android.

1. Tampilan Menu Kamera

Menu kamera adalah menu utama, aplikasi mengakses kamera depan dari perangkat Android untuk melakukan deteksi isyarat secara *real-time*.



Gambar 12 Splash Screen

2. Tampilan Menu Galeri

Pada gambar 4.16 merupakan tampilan menu Galeri, terdapat button plus untuk mengakses galeri perangkat untuk dilakukan deteksi isyarat tidak secara realtime. Gambar berikut merupakan hasil deteksi menggunakan gambar yang diakses dari perangkat.



Gambar 12 Tampilan Hasil Deteksi Gambar

3.7 Testing

Testing merupakan proses penting dalam pengembangan perangkat lunak yang bertujuan untuk mengevaluasi dan memastikan bahwa perangkat lunak sesuai dengan kebutuhan dan harapan yang telah ditentukan. Dalam penelitian ini, digunakan dua pendekatan pengujian, yaitu black box testing dan usability testing.

3.7.1 Blackbox Testing

Blackbox Testing adalah pengujian yang dilakukan dengan cara mencoba program aplikasi dengan memasukkan data ke dalam form form yang telah disediakan. Pengujian ini memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program [12]. Tabel berikut adalah tabel hasil pengujian yang telah dilakukan.

Tabel 1 Hasil Pengujian Black Box

No	Uji Kasus	Skenario Uji Coba	Hasil yang diharapkan	Kesimpulan
1	Deteksi kerangka tangan (hand landmarks)	Deteksi kerangka tangan (hand landmarks)	Sistem mampu mendeteksi semua <i>landmarks</i> tangan dan aplikasi menampilkan titik-titik kerangka di layar Android	Valid
2	Pengujian Deteksi Isyarat Tangan	Pengguna menunjukkan isyarat tangan ke kamera perangkat.	Aplikasi dapat dengan tepat mengenali dan menginterpretasikan setiap isyarat tangan yang ditampilkan	Valid
3	Deteksi isyarat yang bukan huruf	Pengguna menunjukkan tangan dengan simbol yang bukan huruf	Aplikasi memberi respon <i>output</i> bahwa pola yang terbentuk bukan huruf.	Valid
4	Pengujian Responsivitas	Pengguna menunjukkan isyarat tangan dengan cepat.	Aplikasi merespons dengan cepat terhadap gerakan tangan, menampilkan interpretasi secara instan.	Valid

5	Pengujian Kompabilitas Perangkat	Aplikasi dijalankan pada berbagai perangkat Android berbeda	Aplikasi berfungsi dengan baik di berbagai perangkat.	Valid
6	Pengujian Stabilitas	Aplikasi dijalankan selama jangka waktu yang lama.	Aplikasi tetap stabil dan tidak mengalami penurunan kinerja atau kegagalan seiring waktu.	Valid
7	Deteksi dalam kondisi cahaya redup menggunakan MediaPipe	Arahkan tangan dalam kondisi pencahayaan rendah di depan kamera	Aplikasi tetap mendeteksi tangan dengan akurat di kondisi cahaya rendah	Valid
8	Deteksi menggunakan file video dari galeri	Pengguna memilih video dari perangkat	Aplikasi dapat mengakses video dan menampilkan hasil deteksi berdasarkan video yang dipilih	Valid
9	Deteksi menggunakan file gambar dari galeri	Pengguna memilih gambar dari perangkat	Aplikasi dapat mengakses gambar dan menampilkan hasil deteksi berdasarkan gambar yang dipilih	Valid

3.7.2 Usability Testing

Dalam Penelitian ini Pengujian ini dilaksanakan dengan menyebarkan kuesioner digital dan mengumpulkan penilaian dari 60 responden terkait sistem yang telah dikembangkan. Kuesioner tersebut berisi 7 pertanyaan yang dijawab oleh para responden, Berikut hasil perhitungan yang diperoleh:

$$\frac{82,3 + 75 + 85,3 + 80,6 + 87,3 + 87,3 + 85,3 + 85}{8} = 83,51\%$$

Berdasarkan hasil perhitungan rata-rata melalui usability testing, diperoleh nilai presentase sebesar 83,51% sehingga dapat disimpulkan bahwa Sistem “Sistem Deteksi Simbol Pada Sibi (Sistem Isyarat Bahasa Indonesia) Menggunakan Mediapipe Berbasis Android“ sudah memenuhi kriteria interpretasi skor Sangat Setuju.

4. KESIMPULAN

1. proses pengembangan aplikasi deteksi bahasa isyarat berbasis Android menggunakan MediaPipe berjalan dengan baik, mulai dari pembuatan model hingga integrasi ke Android. Hasil pengujian black box menunjukkan aplikasi berhasil mendeteksi semua landmarks tangan, mengenali isyarat secara akurat, merespons dengan cepat, dan tetap stabil di berbagai perangkat serta kondisi cahaya. Selain itu, aplikasi dapat mendeteksi gerakan pada video maupun gambar yang dipilih dengan hasil interpretasi yang jelas dan instan.
2. Model deteksi menunjukkan performa cukup baik dengan akurasi tertinggi 90.85%, namun terdapat tantangan pada beberapa kelas seperti "Q" yang memiliki nilai false negative tinggi. Precision model mencapai 0.88, tetapi model masih kesulitan mendeteksi data dengan pola mirip antar kelas seperti O dan T, R dan U. Secara keseluruhan, model memiliki keseimbangan antara precision dan recall dengan F1-score rata-rata sebesar 0.75.

3. Pengujian dengan 60 pengguna melalui usability testing menunjukkan respons positif dengan rata-rata kepuasan 83.51%. Sebagian besar pengguna menyatakan hasil deteksi jelas dan aplikasi bermanfaat.

5. SARAN

1. Menggunakan dataset video untuk mendeteksi isyarat dinamis seperti huruf J dan Z, sehingga sistem dapat mengenali gerakan dengan lebih akurat dan responsif.
2. Menambahkan kelas untuk kata atau frasa dasar dalam Bahasa Isyarat Indonesia (SIBI) guna meningkatkan cakupan dan kegunaan sistem, mendukung komunikasi, dan pembelajaran bahasa isyarat.
3. Menggunakan jumlah data pada setiap kelas seimbang untuk menghindari bias model terhadap kelas tertentu. Dataset yang tidak seimbang dapat menyebabkan model lebih akurat pada kelas yang dominan, tetapi buruk dalam mengenali kelas dengan data lebih sedikit.

UCAPAN TERIMA KASIH

Penulis menyampaikan rasa terima kasih yang mendalam kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat berjalan dengan lancar. Penghargaan ini terutama ditujukan kepada pembimbing, rekan sejawat, serta instansi terkait yang telah meluangkan waktu, menyediakan sumber daya, dan memberikan bantuan yang sangat berarti selama pelaksanaan penelitian. Semoga segala bentuk kontribusi yang diberikan mendapatkan balasan yang sepadan.

DAFTAR PUSTAKA

- [1] D. D. Putra and H. Gunawan, "Pelaksanaan Pendidikan Tinggi bagi Penyandang Disabilitas di UIN Sulthan Thaha Saifuddin Jambi," *J. Disabil. Stud. Res.*, vol. 1, no. 1, pp. 37–46, 2022.
- [2] M. Sulaeman, "Aksesibilitas Disabilitas Untuk Mewujudkan Pendidikan Yang Inklusif Di Lingkungan Pendidikan Tinggi," vol. 1, no. 5, pp. 65–72, 2024.
- [3] D. P. Lestari and F. Pribadi, "Aksesibilitas dan Sikap Sosial Lingkungan Akademis Mendukung Kegiatan Belajar Mahasiswa Disabilitas," *J. Ilm. Profesi Pendidik.*, vol. 9, no. 2, pp. 975–980, 2024, doi: 10.29303/jipp.v9i2.2212.
- [4] D. M. Vianti, "Efektivitas Penerapan Model Komunikasi Total Bagi Anak Tunarungu Di Bhakti Luhur," *J. Pelayanan Pastor.*, no. April, pp. 46–55, 2021.
- [5] F. N. Rahmah, "Problematika Anak Tunarungu Dan Cara Mengatasinya," *Quality*, vol. 6, no. 1, p. 1, 2018, doi: 10.21043/quality.v6i1.5744.
- [6] M. A. Handayani, "Komunikasi Anak Tuna Rungu Dengan," vol. 3, no. 2, pp. 213–230, 2018.
- [7] I. Sari, Fivrenodi, E. Altirika, and Sarwindah, "Sistem Pengembangan Bahasa Isyarat Untuk Berkomunikasi dengan Penyandang Disabilitas (Tunarungu)," *J. Inf. Technol. Soc.*, vol. 1, no. 1, pp. 20–25, 2023, doi: 10.35438/jits.v1i1.21.
- [8] Adam Nurmansyah, Nanda Rizqia Rhamadhani, Sabrina Alfarissy Nur Hakim, Sri Azhari Agustin, and Siti Hamidah, "Permasalahan Komunikasi Yang Kerap Terjadi Pada Penyandang Disabilitas," *J. Pendidikan, Bhs. dan Budaya*, vol. 2, no. 2, pp. 200–210, 2023, doi: 10.55606/jpbb.v2i2.1515.
- [9] Amanda Muchsin Chalik, Bilal Abdul Qowy, Faiz Hanafi, and Ahlijati Nuraminah, "Mouse Tracking Tangan dengan Klasifikasi Gestur Menggunakan OpenCV dan

- Mediapipe,” *J. Ilm. Tek. Inform. dan Komun.*, vol. 1, no. 2, pp. 10–18, 2021, doi: 10.55606/juitik.v1i2.323.
- [10] M. P. Putri *et al.*, *Algoritma Dan Struktur Data*. 2022.
- [11] Y. Efendi, R. Muzawi, U. Rio, and L. Lusiana, “Aplikasi 3D VR Class Sebagai Inovasi Media Pembelajaran Ditengah Pandemi,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 3, no. 2, pp. 419–424, 2021, doi: 10.47233/jteksis.v3i2.304.
- [12] P. E. Sudjiman and L. S. Sudjiman, “Analisis Sistem Informasi Manajemen Berbasis Komputer Dalam Proses Pengambilan Keputusan,” *TeIKa*, vol. 8, no. 2, pp. 55–66, Jun. 2020, doi: 10.36342/teika.v8i2.2327.
-