

APLIKASI PENGENALAN ABJAD SISTEM ISYARAT BAHASA INDONESIA (SIBI) DENGAN ALGORITMA YOLOv5

MOBILE APPLICATION ALPHABET RECOGNITION OF INDONESIAN LANGUAGE SIGN SYSTEM (SIBI) USING YOLOv5 ALGORITHM

Dimas Permana¹⁾, Joko Sutopo²⁾,

^{1,2}Prodi Informatika, Fakultas Sains dan Teknologi, Universitas Teknologi Yogyakarta

Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I. Yogyakarta 55285

E-mail : ¹* dimaspermana382@gmail.com, ²jksutopo@uty.ac.id

*Corresponding author email.

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan *platform* praktis pengenalan abjad Sistem Isyarat Bahasa Indonesia (SIBI) sebagai media pembelajaran bahasa isyarat yang efektif. Metode yang digunakan adalah algoritma *YOLOv5* untuk mendeteksi isyarat bahasa isyarat SIBI. Pengujian dilakukan dalam dua tahap, pertama untuk menguji antarmuka aplikasi yang berhasil menampilkan enam halaman antarmuka pengguna dengan baik. Kedua, pengujian manual dilakukan untuk mendeteksi gerakan bahasa isyarat SIBI dengan membandingkan kelas-kelas aktual yang terdiri dari 26 kelas. Pada pengujian tersebut, beberapa kelas seperti 'A' dan 'E' memiliki akurasi deteksi yang rendah karena memiliki gerakan yang hampir sama dengan kelas 'S' atau 'A'. Kelas 'D' juga memiliki akurasi yang rendah karena membutuhkan variasi posisi yang lebih kompleks. Kelas 'J' memiliki gestur yang cukup rumit karena ada gerakan lanjutan yang dilakukan, sehingga memiliki akurasi yang cukup rendah. Hasil deteksi real-time menggunakan kamera ponsel menunjukkan akurasi sebesar 77% (0.77), secara keseluruhan hasil pengujian mencapai akurasi yang cukup baik. Diharapkan dengan adanya platform ini dapat mempermudah dan memperlancar komunikasi antara masyarakat umum dengan teman tuli.

Kata kunci : Bahasa Isyarat, Deteksi Objek, Komunikasi Tuli, Sibi, Yolo.

ABSTRACT

This research aims to develop a practical platform of Indonesian Sign Language System (SIBI) alphabet recognition as an effective sign language learning media. The method used is YOLOv5 algorithm based on convolutional neural network (CNN) to detect SIBI sign language gestures. Testing is done in two stages, first to test the application interface which successfully displays 6 user interface pages properly. Secondly, manual testing was carried out to detect SIBI sign language gestures by comparing the actual classes consisting of 26 classes. In the test, some classes such as 'A' and 'E' had low detection accuracy due to almost the same gesture as class 'S' or 'A'. Class 'D' also has low accuracy as it requires more complex position variations. Class 'J' is difficult to detect due to the advanced movements required. The results of real-time detection using a cell phone camera showed an accuracy of 77% (0.77), overall the test results achieved a fairly good accuracy. It is hoped that this platform can facilitate easy and smooth communication between the general public and deaf friends.

Keywords: Deaf Communication, Object Detection, Sibi, Sign Language, Yolo.

PENDAHULUAN

Tunarungu adalah kondisi medis di mana seseorang kehilangan kemampuan untuk mendengar suara atau bunyi. Tuli dapat terjadi secara sebagian atau total, dan dapat terjadi sejak lahir atau terjadi akibat kondisi kesehatan atau kecelakaan. Beberapa faktor risiko yang dapat menyebabkan tuli meliputi infeksi telinga yang berulang, paparan suara yang berlebihan, cedera kepala, atau faktor genetik. Kondisi ini dapat mengganggu kemampuan seseorang untuk berkomunikasi dan berinteraksi dengan lingkungannya, dan dapat memengaruhi kualitas hidup seseorang secara keseluruhan.

Menurut data dari Kementerian Sosial Republik Indonesia pada tahun 2020, terdapat sekitar 1,5 juta orang penyandang tunarungu atau tuli di Indonesia. Angka ini menunjukkan bahwa tuli adalah kondisi yang relatif umum di masyarakat Indonesia. Meskipun demikian, akses terhadap alat bantu pendengaran dan terapi pendengaran masih terbatas, terutama di daerah pedesaan dan kawasan miskin. Hal ini menyebabkan banyak penyandang tunarungu kesulitan dalam berkomunikasi dan berpartisipasi dalam kegiatan sosial dan ekonomi.

Cara teman tuli berkomunikasi dengan menggunakan bahasa isyarat. Bahasa isyarat ini menggunakan gerak tangan, ekspresi wajah atau tubuh untuk membentuk simbol-simbol yang mewakili sebuah huruf atau kata [1]. Di Indonesia ada dua bahasa isyarat yang populer digunakan, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Namun bahasa isyarat sibi lebih dianggap sebuah sistem pembelajaran, sedangkan bisindo tercipta karena interaksi sejak kecil [2].

Proses komunikasi masyarakat awam dengan teman tuli dirasa kurang efektif jika menggunakan media tambahan berupa penulisan pada kertas atau media lainnya, sehingga membutuhkan waktu yang lama untuk

menulis, kemudian membacakan dan kemudian menulis balasan jawabannya.

Pada umumnya jika tanpa media penulisan teman-teman tuli menggunakan seorang penerjemah antara orang biasa dengan mereka. Penerjemah ini adalah orang yang paham dan dapat berkomunikasi menggunakan bahasa isyarat dengan lancar, namun hal ini merupakan sesuatu yang membuang-buang waktu dan membutuhkan biaya tambahan untuk menyewa penerjemah tersebut.

Saat ini penggunaan *smartphone* yang berkembang pesat saat ini dapat dijadikan target dalam mengembangkan aplikasi khusus penerjemah bahasa isyarat. *Smartphone* ini dinilai cara yang baik digunakan untuk mengimplementasikan aplikasi mengingat masyarakat Indonesia hampir semuanya memiliki *smartphone* dan sudah dapat mendukung dan menjalankan sistem terjemah bahasa isyarat dengan deteksi gerakan. Pengenalan bahasa isyarat menggunakan kamera dan sistem bertindak sebagai pelacakan visual dari setiap gerakan dan ekspresi tangan penyandang tunarungu.

Beberapa aplikasi telah dikembangkan sebagai upaya untuk membantu menerjemahkan bahasa isyarat ke menjadi teks abjad. Pada penelitian bahasa isyarat ini objek tangan merupakan hal yang utama untuk dibahas. Pemilihan metode CNN (*Convolutional Neural Network*) untuk proses klasifikasi deteksi dipilih untuk mendeteksi bahasa isyarat Indonesia[3].

Metode YOLO V3 digunakan dalam penelitian ini mendeteksi *gesture* tangan bahasa isyarat, dalam prosesnya *Yolo* ini dapat melakukan proses *tracking* walaupun kondisi intensitas cahaya kurang sehingga hasil yang didapatkan cukup baik dalam deteksi isyarat [4]. Aplikasi android untuk menerjemahkan bahasa isyarat menjadi teks sekaligus hasil dari terjemahan dapat dikonversi menjadi suara (*text to speech*). Hal ini akan memudahkan dalam proses berkomunikasi antara teman tuli dan teman dengar. Metode yang digunakan

CNN dan memakai model *MobileNetV2* [5]. Penelitian ini menggunakan model bahasa isyarat *Amerika Sign Language* untuk datasetnya. Kemudian *dataset* tersebut dilatih dengan metode CNN (*Convolutional Neural Network*). Dari penelitian ini hasil akurasi yang didapatkan adalah 82,1% akurasi yang cukup baik dalam proses deteksi objek, tetapi model ASL tidak umum digunakan di Indonesia sehingga perlu adanya pembaharuan model yang dilatih menggunakan BISINDO atau SIBI [6]. Penelitian ini [7] telah berhasil membuat aplikasi android untuk menerjemahkan bahasa isyarat menjadi teks. Metode yang digunakan berupa CNN (*Convolutional Neural Network*) dan library *Tensorflow Lite*, selain itu peneliti menambahkan beberapa fitur unggulan seperti Deteksi Isyarat, dan Dictionary sehingga pengguna tidak hanya bisa menerjemahkan bahasa isyarat saja tapi juga dapat belajar bahasa isyarat dari gambar yang telah disediakan. Penelitian lainnya telah dilakukan klasifikasi *human hand detection* menggunakan *three-dimensional hand pose estimation*. Sistem dapat mendeteksi gerakan posisi tangan manusia, didapatkan hasil akurasi rata-rata 96,2% [8].

Berdasarkan kajian pustaka dengan penelitian terdahulu, maka tujuan penelitian ini adalah untuk membuat platform pengenalan abjad sibi yang praktis dan media pembelajaran bahasa isyarat yang efektif. Sehingga dalam hal ini masyarakat awam dan teman tuli dapat berkomunikasi dengan mudah tanpa adanya hambatan.

METODE

Metode penelitian diawali dengan studi literatur dengan *review paper* penelitian sebelumnya yang berkaitan dengan topik penelitian. Kemudian dilanjutkan dengan tahap analisis kebutuhan sistem, tahap perancangan sistem, tahap implementasi sistem, dan terakhir pengujian aplikasi. Metode yang digunakan untuk proses pengenalan adalah *You Only Look Once (YOLOv5)*.

Gambaran alur penelitian ditunjukkan pada gambar 1 dibawah ini.



Gambar 1. Desain 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.

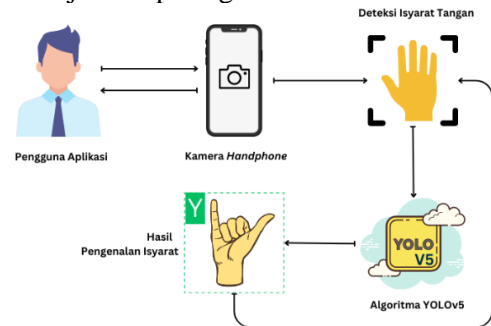
Perancangan sistem meliputi proses pelatihan *model machine learning*, pembuatan data citra, dan desain antarmuka aplikasi. Untuk implementasi *model* akan dimasukkan kedalam sistem *mobile android* dan setelah itu tahap terakhir adalah pengujian aplikasi kepada pengguna.

A. Arsitektur Sistem Aplikasi

1. Arsitektur Sistem

Arsitektur sistem, menjelaskan proses awal dari Pengguna mengakses aplikasi kemudian mengarahkan kamera ke objek tangan teman tuli yang melakukan gerakan bahasa isyarat. Proses

klasifikasi objek akan dilakukan oleh algoritma *YOLO v5*, jika sistem dapat mengenali maka akan menampilkan *bounding box* dan *output* berupa teks huruf abjad. Proses ini akan terjadi iterasi sesuai apa yang dideteksi oleh kamera *handphone*. Gambaran arsitektur sistem ditunjukkan pada gambar 2 dibawah ini.



Gambar 2. Arsitektur Sistem.

Pada gambar 2 merupakan proses berjalannya sistem yang akan dilakukan. Pengguna aplikasi diharuskan mengarahkan kamera *handphone* nya kepada teman tuli yang melakukan peragaan Bahasa isyarat abjad. Jika sistem berhasil mengenali objek maka pada aplikasi akan menampilkan terjemahan berupa teks. Proses pengenalan tersebut akan berlangsung terus-menerus apabila diarahkan pada objek tangan.

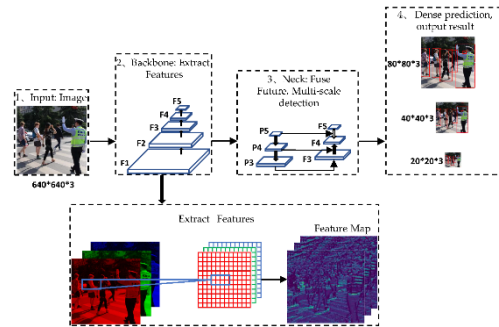
2. Algoritma *YOLOv5*

You Only Look Once (YOLO) adalah metode deteksi objek yang dikenal karena kecepatannya [9]. Diperlukan hanya satu jaringan konvolusi yang digunakan untuk memprediksi objek apa yang akan ada dalam gambar. Hal ini dapat terjadi karena *YOLO* membagi gambar menjadi sel/grid, setiap sel bertanggung jawab untuk memprediksi jumlah kotak pembatas (*bounding box*), tingkat kepercayaan setiap sel, dan probabilitas kelas.

Algoritma *YOLOv5* memakai pendekatan deteksi objek berbasis jaringan saraf konvolusi (CNN). CNN adalah jenis arsitektur jaringan saraf yang sangat baik dalam mengenali pola dan fitur dalam data gambar. *YOLOv5* menggunakan arsitektur *EfficientNet* sebagai dasar dari modelnya. *EfficientNet* adalah sebuah arsitektur CNN yang

efisien dan sangat kuat dalam pengenalan gambar.

Salah satu fitur utama dari *YOLOv5* adalah kemampuannya untuk mendeteksi objek secara real-time pada perangkat dengan daya komputasi yang terbatas. Ini berarti *YOLOv5* dapat berjalan dengan cepat dan efisien bahkan pada perangkat dengan sumber daya terbatas seperti perangkat *mobile* atau *embedded systems*.



Gambar 3. Arsitektur *YOLOv5*.

Pada gambar 3 merupakan arsitektur dari *YOLOv5*. Cara kerja *YOLOv5* dengan mengubah gambar input menjadi grid yang terdiri dari sel-sel kecil[10]. Setiap sel bertanggung jawab untuk mendeteksi objek yang berada di dalamnya. Model CNN kemudian melihat seluruh gambar secara keseluruhan dan menghasilkan prediksi probabilitas untuk kelas objek yang berbeda serta kotak pembatas (*bounding box*) yang mengelilingi objek [11].

YOLOv5 mengimplementasikan beberapa teknik untuk meningkatkan akurasi dan kecepatan deteksi, seperti metode augmentasi data untuk memperluas dataset pelatihan, penggunaan teknik pemisahan objek untuk meningkatkan kemampuan deteksi, dan metode pengurangan bobot untuk mempercepat proses pelatihan.

YOLOv5 juga dapat mengatasi beberapa tantangan dalam deteksi objek, seperti objek yang tumpang tindih, objek dengan ukuran yang berbeda-beda, dan objek dalam kondisi pencahayaan yang berbeda. Dengan kata lain, algoritma *YOLOv5* adalah sebuah metode deteksi objek yang efisien dan akurat, yang memungkinkan deteksi objek secara real-time pada perangkat dengan sumber daya

terbatas. Algoritma ini telah digunakan dalam berbagai aplikasi seperti deteksi objek di video, kendaraan otonom, sistem keamanan, dan banyak lagi.

3. Object Detection

Deteksi objek adalah teknologi komputer yang terkait dengan penglihatan komputer dan pengolahan citra yang berkaitan dengan mendeteksi objek dalam gambar *digital* berdasarkan warna dan bentuk objek [12].

Deteksi objek menggunakan metode *YOLOv5* memungkinkan identifikasi objek dalam gambar digital atau objek didalam video. *Object detection* dapat dibagi menjadi *soft detection* yang hanya mendeteksi keberadaan dari objek dan *hard detection* yang mendeteksi keberadaan serta lokasi dari objek [13]. Dalam penelitian ini model *object detection* yang dilatih yaitu untuk mendeteksi sistem isyarat bahasa Indonesia (sibi).

4. Tensorflow Lite

TensorFlow Lite adalah sebuah *framework deep learning open source* yang dirancang khusus untuk pelatihan model [14]. *TensorFlow Lite* merupakan versi ringan dari framework TensorFlow yang dirancang khusus untuk menjalankan model *machine learning* pada perangkat dengan sumber daya terbatas, seperti ponsel dan perangkat IoT.

Proses untuk konversi dari *YOLOv5* ke *Tensorflow lite* cukup mudah. Langkah pertama adalah mengonversi model *YOLOv5* yang telah dilatih ke dalam format model *TensorFlow Lite*. Proses konversi ini melibatkan kompresi dan optimasi model untuk mengurangi ukuran dan meningkatkan kecepatan *inference* dan tetap mempertahankan akurasi yang tinggi.

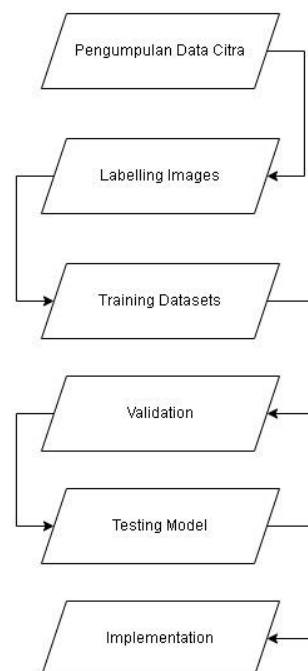
Setelah model *YOLOv5* dikonversi ke format *TensorFlow Lite*, model tersebut dapat diimplementasikan dan dieksekusi pada perangkat mobile menggunakan runtime *TensorFlow Lite*. Runtime *TensorFlow Lite* dirancang untuk menjadi cepat dan efisien, dengan memanfaatkan kemampuan akselerasi perangkat keras seperti GPU, DSP, atau *Neural Processing Units (NPU)* yang ada

pada perangkat target. Hal ini memungkinkan deteksi objek secara real-time atau mendekati real-time menggunakan *YOLOv5* pada perangkat tinggi dengan sumber daya komputasi terbatas.

TensorFlow Lite juga menyediakan kumpulan *API* dan alat untuk pengembang guna mengintegrasikan model *YOLOv5* ke dalam aplikasi. Ini termasuk *API inference* yang telah dibangun sebelumnya untuk melakukan deteksi objek menggunakan model *YOLO*, serta alat untuk modifikasi dan optimisasi model[15].

Kolaborasi antara *TensorFlow lite* dengan *YOLOv5* memungkinkan deteksi objek di perangkat, menghilangkan kebutuhan akan koneksi internet yang konstan atau mengandalkan *inference* berbasis *cloud*. Ini sangat berguna untuk aplikasi yang membutuhkan deteksi objek secara real-time atau beroperasi di lingkungan dengan konektivitas terbatas.

B. Proses Training Datasets



Gambar 4. 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. Pengumpulan Data Citra

Pengumpulan dataset dilakukan dengan pengambilan *sample* gambar oleh penulis sendiri. Gambar citra yang diambil berjumlah 259 citra digital peragaan bahasa isyarat sibi abjad A sampai Z dengan format .jpg dan 3 video peragaan abjad dari A-Z untuk *testing datasets*. Referensi peragaan bahasa isyarat diambil dari website resmi Lembaga Pengelola Sistem Isyarat Bahasa Indonesia. Contoh kumpulan data citra untuk dilatih pada gambar 5 dibawah.

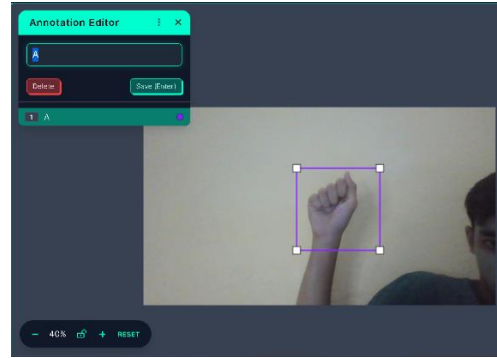


Gambar 5. Data Citra Sibi

Pada gambar 5 merupakan kumpulan data citra yang diambil secara mandiri. Posisi memperagakan bahasa isyarat dilakukan dengan berbagai posisi dan jarak dekat maupun jauh. Data citra berjumlah 26 *class*.

2. Label Images

Proses *annotation* atau pemberian label dilakukan untuk memberikan panduan *machine learning* agar sistem dapat melatih dirinya untuk mendeteksi posisi isyarat tangan saja. Pembelian label pada citra dilakukan dengan website <https://www.roboflow.com> ini merupakan website *powerful* untuk melakukan pelatihan model *machine learning* selain gratis web ini juga mudah untuk digunakan.

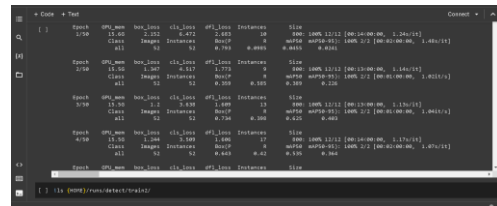


Gambar 6 Labelling Image

Pada gambar 6 merupakan proses memberikan label pada data citra. Pada tahap ini dilakukan secara berulang pada setiap citra sesuai dengan isyaratnya. Jumlah *class* citra yang dihasilkan 26 *class* masing-masing memiliki 10 gambar.

3. Training Data Citra

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 *popular* dan ringan digunakan.



Gambar 7. Proses Training Datasets.

Pada gambar 7 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 *best.pt* atau (*PyTorch*).

Proses selanjutnya yaitu *validate model* untuk mengecek apakah semua *class* sudah dilatih atau belum. Kemudian selanjutnya *testing model* menggunakan citra yang sudah dibagi pada proses awal pelatihan.



Gambar 8. *Testing Model dengan Image.*

Pada gambar 8 merupakan hasil dari *test model training*. Proses menandakan bahwa *model* yang dihasilkan sudah sesuai dan bisa mendeteksi bahasa isyarat yang sudah dilatih. Selain itu hasil dari *test* menampilkan *bounding box* dan nilai deteksi dengan sangat baik dan posisinya tepat pada tangan peraga.

Testing model juga dilakukan dengan menggunakan *video* yang diambil mandiri oleh penulis. *Video* yang dijadikan testing berformat *.mp4* dan berdurasi 1.30 detik.



Gambar 9. *Testing Model dengan Video.*

Pada gambar 9 menampilkan hasil dari *testing model* dengan *video* peragaan bahasa isyarat sibi. *Video* menampilkan *bounding box* yang tepat berada pada tangan peraga. Jika posisi tangan digerakan maka *bounding box* akan mengikuti dan terus mendeteksi. Sehingga dalam hal ini proses pelatihan *model* dikatakan berhasil.

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 penerjemah komunikasi antara teman dengar dan teman tuli. Selain itu juga memiliki fitur sebagai media pembelajaran bahasa isyarat sibi dan juga menampilkan artikel berkaitan dengan bahasa isyarat, teman tuli dan sejenisnya.

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	Sistem dapat menampilkan halaman pengenalan aplikasi.
2	Sistem dapat menampilkan kumpulan peraga bahasa isyarat (kamus sibi).
3	Sistem dapat menampilkan informasi artikel berkaitan dengan bahasa isyarat dan tuli.
4	Sistem dapat menampilkan hasil pengenalan pada aplikasi.
5	Sistem dapat mengakses kamera dari pengguna.

Pada Tabel 1 menampilkan kebutuhan fungsional aplikasi. Dalam prosesnya memiliki tiga fitur utama yaitu fitur terjemah bahasa isyarat, fitur kamus bahasa isyarat, dan fitur artikel informasi.

3. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang memerlukan properti untuk mendukung sistem sehingga dapat berjalan dengan baik. Kebutuhan non-fungsional dapat dilihat pada Tabel 2.

Tabel 2. Kebutuhan Non-fungsional Aplikasi.

No	Kebutuhan Non-fungsional
1	Sistem dapat berjalan pada sistem ponsel terbatas dengan minim spesifikasi.
2	Sistem dapat digunakan dengan mudah oleh pengguna awam.

Pada tabel 2 merupakan kebutuhan non-fungsional aplikasi. Dalam hal ini

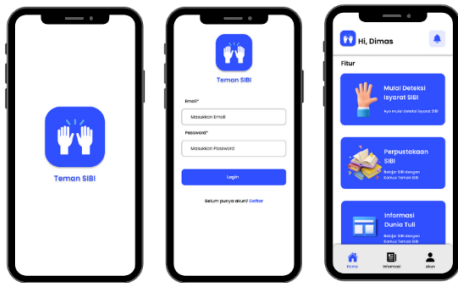
peneliti mempertimbangkan bahwa sistem atau aplikasi harus dapat dijalankan diperangkat yang memiliki spesifikasi minimum sehingga dapat diakses oleh siapapun. Kemudian kebutuhan lainnya yaitu sistem dapat digunakan dengan mudah oleh siapapun sehingga disini dibutuhkan analisis pada desain antarmuka yang baik.

HASIL DAN PEMBAHASAN

A. Implementasi Antarmuka

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

Aplikasi ini memiliki 6 antarmuka yang terdiri dari halaman *splash screen*, halaman utama, halaman terjemah, halaman kamus, halaman artikel informasi, halaman *login* pengguna, dan halaman akun. Berikut pada gambar 10 dan 11 contoh gambaran dari antarmuka aplikasi.



Gambar 10. Implementasi Antarmuka Aplikasi 1.



Gambar 11. Implementasi Antarmuka Aplikasi 2.

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

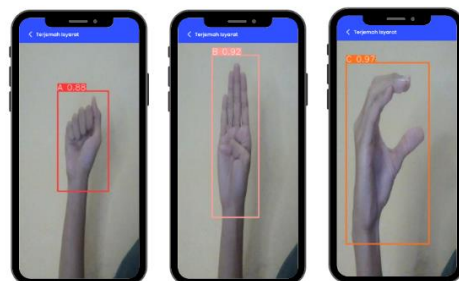
B. Pengujian Antarmuka Pengguna
 Pengujian antarmuka terdiri dari pengujian fitur aplikasi yang dilakukan dengan metode *blackbox*. Berikut merupakan Tabel 3 hasil pengujian antarmuka pada aplikasi.

Tabel 3. Hasil Pengujian Antarmuka.

Aktifitas Pengujian	Hasil Pengujian	Keterangan
Membuka aplikasi	Menampilkan halaman <i>login</i>	Berhasil
<i>Login</i> memasukkan email dan <i>password</i>	Menampilkan halaman utama	Berhasil
Membuka menu terjemah	Menampilkan halaman terjemah bahasa isyarat	Berhasil
Membuka menu kamus	Menampilkan halaman kamus Bahasa isyarat	Berhasil
Membuka menu artikel	Menampilkan halaman artikel informasi	Berhasil
Keluar aplikasi	Menutup aplikasi dan Kembali ke halaman ponsel	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 12 merupakan contoh hasil deteksi yang berhasil.



Gambar 12. 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.

Dalam pengujian akurasi didapatkan dari hasil deteksi selama 10 kali pada masing-masing *class*. Sehingga hasil didapatkan seperti pada Tabel 4 dibawah.

Tabel 4. Hasil Pengujian Akurasi.

La bel	Jumlah Tahap Pengujian										Aku rasi
	1	2	3	4	5	6	7	8	9	10	
A	X	X	X	√	√	√	X	X	X	X	0,30
B	X	√	√	√	√	√	√	√	√	√	0,90
C	√	√	√	√	√	√	√	√	√	√	1,00
D	X	X	X	X	√	X	X	X	X	X	0,10
E	X	X	X	X	X	X	X	X	X	X	0,00
F	√	√	X	√	X	X	X	√	√	√	0,60
G	√	√	√	√	√	√	√	√	√	√	1,00
H	√	√	√	√	√	√	√	√	√	√	1,00
I	√	√	√	√	√	√	√	√	√	√	1,00
J	X	√	X	X	X	X	X	X	X	X	0,10
K	X	√	√	√	√	√	√	√	√	√	0,90
L	√	√	√	√	√	√	√	√	√	√	1,00
M	√	√	√	√	√	√	√	√	√	√	1,00
N	X	X	√	√	√	√	√	√	√	√	0,80
O	X	√	X	√	√	√	√	√	√	√	0,80
P	√	√	X	√	√	√	√	√	√	√	0,90
Q	√	√	√	√	√	√	√	√	√	√	1,00
R	X	√	√	√	√	√	√	√	√	√	0,90
S	√	X	X	√	√	√	√	√	√	√	0,80
T	X	√	√	X	√	√	√	√	√	X	0,70
U	X	X	√	√	√	√	√	√	√	√	0,80
V	√	√	√	√	√	√	√	√	√	√	1,00
W	√	√	√	√	√	√	√	√	√	√	1,00
X	X	X	√	√	X	√	√	√	√	√	0,70
Y	√	√	√	√	√	√	√	√	√	√	1,00
Z	√	X	√	√	√	X	√	√	√	√	0,80
Nilai Rata-rata Akurasi											0,77

Pengujian kelas ‘A’ terdeteksi akurasi rendah dikarenakan gesturnya hampir sama dengan kelas ‘S’. Kelas ‘D’ terdeteksi rendah dikarenakan bentuk peraga huruf ‘D’ memerlukan posisi dari berbagai arah. Kelas ‘E’ tidak terdeteksi sama sekali dikarenakan hampir sama gesturnya dengan kelas ‘S’ dan ‘A’. Kelas ‘J’ sulit terdeteksi dikarenakan gesturnya yang membutuhkan gerakan lanjutan.

Berdasarkan tabel pengujian diatas dapat disimpulkan bahwa hasil pengujian mendapatkan akurasi untuk 26 *class*

adalah 77% (0,77). Hasil akurasi dapat dikatakan cukup baik dan berhasil.

SIMPULAN

Arsitektur algoritma *YOLOv5* menggunakan pendekatan deteksi objek berbasis jaringan saraf konvolusi (CNN). CNN adalah jenis arsitektur jaringan saraf yang sangat baik dalam mengenali pola dan fitur dalam data gambar. *YOLOv5* menggunakan arsitektur *EfficientNet* sebagai dasar dari modelnya. *EfficientNet* adalah sebuah arsitektur CNN yang efisien dan sangat kuat dalam pengenalan gambar. Proses training dengan algoritma *YOLOv5* dilakukan pada 300 data cita bahasa isyarat sibi.

Pengujian dilakukan dengan dua tahap, pertama untuk pengujian antarmuka dari aplikasi yang memiliki 6 halaman antarmuka pengguna yang semuanya berhasil ditampilkan dengan baik.

Pengujian kedua dilakukan dengan manual untuk mendeteksi gerakan bahasa isyarat sibi. Proses pengujian dengan membandingkan kelas sebenarnya berjumlah 26 *class* dengan hasil deteksi *real-time* menggunakan kamera ponsel. Didapatkan hasil akurasi 77% (0,77).

SARAN

Penelitian ini masih memiliki banyak kekurangan, sehingga peneliti memiliki beberapa saran untuk peneliti yang ingin mengembangkan sistem aplikasi ini dimasa yang akan datang. Diantaranya sebagai berikut.

Pertama, menambahkan jumlah data citra untuk dilatih dengan algoritma *YOLOv5*. Hal tersebut untuk mendapatkan pengalaman akurasi yang lebih baik lagi diperlukan banyak datasets dengan berbagai kondisi.

Kedua, mengembangkan fitur-fitur yang sudah ada maupun menambahkan fitur baru yang membuat aplikasi ini dapat dirasakan pengalaman berkomunikasi dengan teman tuli menjadi lebih baik lagi.

DAFTAR PUSTAKA

- [1] I. Inayatul Arifah, F. Nur Fajri, and G. Qorik Oktagal Pratamasunu, “Deteksi

- Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN,” 2022. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [2] M. Zikky, Z. F. Akbar, and S. Utomo, “Kamus sistem isyarat bahasa Indonesia (KASIBI) dengan voice recognition sebagai pendukung belajar bahasa isyarat berbasis android,” *JST (Jurnal Sains Terapan)*, vol. 5, no. 2, pp. 121–130, 2019.
- [3] M. Yunus and Y. Anwar, “Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke dalam Huruf Abjad,” 2022. [Online]. Available: <https://jurnal.umpar.ac.id/index.php/sylog/257>
- [4] A. Sani and S. Rahmadinni, “Deteksi Gestur Tangan Berbasis Pengolahan Citra,” *Jurnal Rekayasa ElektriKa*, vol. 18, no. 2, Jul. 2022, doi: 10.17529/jre.v18i2.25147.
- [5] N. Hikmatia and M. I. Zul, “Aplikasi Penerjemah Bahasa Isyarat Indonesia Menjadi Suara berbasis Andoid menggunakan Tensorflow,” 2021. [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jkt/>
- [6] F. D. Siswanto, C. C. Lestari, and E. Tanuwijaya, “Klasifikasi Bahasa Isyarat Amerika menggunakan Convolutional Neural Network,” vol. 10, no. 1, 2022, doi: 10.26418/justin.v10i1.47184.
- [7] 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>
- [8] H. C. Nguyen, T. H. Nguyen, R. Scherer, and V. H. Le, “YOLO Series for Human Hand Action Detection and Classification from Egocentric Videos,” *Sensors (Basel)*, vol. 23, no. 6, Mar. 2023, doi: 10.3390/s23063255.
- [9] S. Daniels, N. Suciati, and C. Fathichah, “Indonesian Sign Language Recognition using YOLO Method,” *IOP Conf Ser Mater Sci Eng*, vol. 1077, no. 1, p. 012029, Feb. 2021, doi: 10.1088/1757-899x/1077/1/012029.
- [10] F. Sun, “Face Recognition Analysis Based on the YOLO Algorithm,” *Applied and Computational Engineering*, vol. 2, no. 1, pp. 213–222, Mar. 2023, doi: 10.54254/2755-2721/2/20220679.
- [11] M. Bhavadharshini, J. Josephine Racheal, M. Kamali, S. Sankar, and M. Bhavadharshini, “Sign language translator using YOLO algorithm,” *Advances in Parallel Computing*, vol. 39, pp. 159–166, 2021, doi: 10.3233/APC210136.
- [12] R. Hesandana, D. Natasya, and N. Wiliani, “Cloth BAG Object Detection Using The Yolo Algorithm (You Only See Once) V5,” *Jurnal Pilar Nusa Mandiri*, vol. 18, no. 2, pp. 217–222, Feb. 2023, doi: 10.33480/pilar.v18i2.3019.
- [13] S. S. Sindarto, D. E. Ratnawati, and I. Arwani, “Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) dengan Metode Convolutional Neural Network pada Perangkat Lunak berbasis Android,” vol. 6, no. 5, pp. 2129–2138, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [14] U. Fadlilah, A. K. Mahamad, and B. Handaga, “The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An Initial Study,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Apr. 2021. doi: 10.1088/1742-6596/1858/1/012085.
- [15] Y. Li, Z. Zhao, Y. Luo, and Z. Qiu, “Real-time pattern-recognition of GPR images with YOLO V3 implemented by tensorflow,” *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–17, Nov. 2020, doi: 10.3390/s20226476.