



**APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS REACT JS**

TUGAS AKHIR

KHOIR AFANDI

NPM 20670121

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024



**APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS REACT JS**

TUGAS AKHIR

KHOIR AFANDI

NPM 20670121

Diajukan kepada Fakultas Teknik dan Informatika

Universitas PGRI Semarang guna

Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS PGRI SEMARANG

2024

TUGAS AKHIR
APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS REACT JS

Diajukan dan disusun oleh

KHOIR AFANDI

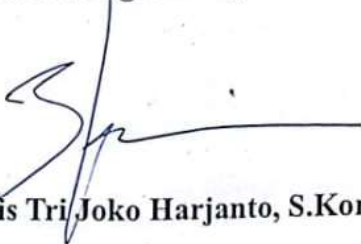
NPM 20670121

telah disetujui pembimbing guna dilanjutkan untuk

disusun menjadi skripsi

Pada tanggal 2 Juli 2024

Pembimbing Utama,



Aris Tri Joko Harjanto, S.Kom., M.Kom.
NIDN. 0619048202

Pembimbing Pendamping,



Febrian Murti Dewanto, SE., M. Kom.
NIDN. 0626028201

TUGAS AKHIR

**APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS REACT JS**

Disusun dan diajukan oleh

KHOIR AFANDI

NPM 20670121

Telah dipertahankan di depan Dewan Penguji

pada tanggal 08 Juli 2024

dan dinyatakan telah memenuhi syarat Dewan Penguji



Ketua,

Hani Foto Husodo, S.T., M.T.

NPP. 136901387

Sekretaris,

Bambang Agus H., S.Kom., M.Kom

NIDN. 0601088201

Penguji I,

Aris Tri Joko Harjanto, S.Kom., M.Kom.

NIDN. 0619048202

Penguji II,

Febrian Murti Dewanto, SE., M. Kom.

NIDN. 0626028201

Penguji III,

Bambang Agus H., S.Kom., M.Kom

NIDN. 0601088201

MOTTO DAN PERSEMBAHAN

“Tidak semua hal yang ada di dunia bisa kita cegah, itu namanya takdir. Tuhan tidak pernah memberikan masalah tanpa jalan keluar, kita hanya perlu menghadapinya saja”

- Diosetta

“kesuksesan adalah hasil dari kebiasaan sehari-hari, bukan transformasi yang terjadi sekali seumur hidup”

- Atomic Habits

Persembahan:

Kupersembahkan skripsi ini untuk:

1. Kedua orang tuaku tercinta.
2. Sahabat dan Teman-temanku yang selalu menyemangatiku mengerjakan skripsi.
3. Almamaterku Universitas PGRI Semarang.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Khoir Afandi
NPM : 20670121
Program Studi : Informatika
Fakultas : Teknik dan Informatika
Instansi : Universitas PGRI Semarang

Menyatakan dengan ini keaslian dan kebenaran terkait tugas akhir yang saya susun serta merupakan hasil karya sendiri, bukan plagiarisme.

Apabila pada kemudian hasil proposal skripsi ini terbukti hasil plagiarisme, saya bersedia atas sanksi atas tindakan saya tersebut

Semarang ,01 juli 2024

Pembuat Pernyataan



Khoir Afandi

NPM. 20670121

ABSTRAK

Komunikasi merupakan hal yang sangat penting bagi manusia untuk berinteraksi dan menyampaikan informasi. Namun, bagi penyandang tuna rungu dan tuna wicara, komunikasi menjadi tantangan karena mereka menggunakan bahasa isyarat sebagai alat komunikasi. Bahasa Isyarat Indonesia (BISINDO) adalah bahasa isyarat yang umum digunakan di Indonesia, tetapi banyak orang normal yang tidak memahaminya, sehingga menyulitkan komunikasi antara kedua kelompok ini. Penelitian ini mengembangkan sebuah aplikasi pengenalan huruf abjad bahasa isyarat menggunakan metode Convolutional Neural Network (CNN) yang bertujuan untuk memudahkan komunikasi antara penyandang tuna rungu dan tuna wicara dengan orang normal. Penelitian ini membatasi fokus pada penerjemahan dan pengenala huruf abjad dalam Bahasa Isyarat Indonesia (BISINDO) menggunakan metode CNN berbasis *framework React JS* untuk antarmuka pengguna. Dengan menggunakan dataset gambar bahasa isyarat, CNN dilatih untuk mengenali pola isyarat dan menerjemahkannya ke dalam huruf yang sesuai. Hasil proses pelatihan model untuk pengenalan huruf abjad Bahasa Isyarat BISINDO menggunakan metode CNN menghasilkan akurasi sebesar 96%, dan akurasi validasi sebesar 90%. Pelatihan model ini menggunakan teknik *transfer learning MobileNetV2* dengan dataset training sebanyak 7.800 gambar (80%) dan validasi sebanyak 1.950 gambar (20%). Faktor-faktor seperti pencahayaan, dan variasi gestur memang mempengaruhi akurasi pengenalan, namun secara keseluruhan sistem ini mampu memberikan kontribusi signifikan dalam meningkatkan aksesibilitas komunikasi bagi penyandang tuna rungu dan tuna wicara. Penelitian ini diharapkan dapat mengurangi kesenjangan komunikasi dan menjadi landasan untuk inovasi masa depan dalam teknologi bantu bagi disabilitas. Dalam pengujian pada aplikasi pengenalan huruf abjad bahasa isyarat menggunakan metode Convolutional Neural Network (CNN) ini menggunakan 3 pengujian *black box*, *white box* dan *UAT (User Acceptance Testing)*. Pada *black box* dilakukan pada 3 dosen informatika menghasilkan keberhasilan 100%, *white box* menghasilkan hasil 100% keberhasilan dan mendapatkan 2 *independent path*, dan yang terakhir pengujian UAT menghasilkan 95.2% dari 5 responden dengan 10 pertanyaan.

Kata kunci: *Convolutional Neural Network*, Bahasa Isyarat Indonesia, *MobileNetV2*, *Transfer Learning*, Aplikasi Penerjemah *black box*, *white box* dan *UAT (User Acceptance Testing)*.

PRAKATA

Puji dan syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Karunia -Nya sehingga penyusunan Tugas Akhir yang berjudul “Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Mengunaka Metode Convolutional Neural Network (CNN) Berbasis React JS” ini dengan tepat waktu. Selesaiannya skripsi ini tidak lepas dari doa, bantuan, dukungan serta bimbingan dari beberapa pihak. Oleh karena itu, dalam kesempatan ini dengan tulus hati penulis menyampaikan terima kasih kepada:

1. Allah SWT Yang Maha Pemurah Lagi Maha Penyayang, yang telah memberikan kesempatan dan kemudahan dalam menyelesaikan proposal skripsi dengan lancar.
2. Dr. Sri Suciati, M. Hum.I, selaku Rektor Universitas PGRI Semarang yang telah memberi kesempatan kepada penulis untuk menimba ilmu di Universitas PGRI Semarang.
3. Bapak Ibnu Toto Husodo, S.T., M.T. selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang.
4. Bambang Agus Herlambang, S.Kom., M.Kom., selaku Ketua Program Studi Informatika.
5. Bapak Aris Tri Jaka Harjanta, S.Kom., M.Kom., selaku Pembimbing I telah menyetujui sekaligus memberikan arahan, bimbingan, dan dukungan.
6. Febrian Murti Dewanto, SE., M. Kom., selaku Pembimbing Kedua yang telah membimbing penulis dengan penuh dedikasi dan kesabaran.
7. Bapak dan Ibu Dosen terkhusus di Pogram Studi Informatika Universitas yang memberikan ilmu, dedikasi dan membimbing penulis samapai pada di titik ini.
8. Kedua orang tua, dan seluruh keluarga yang senantiasa memberikan kasih sayang dan cintanya serta selalu mendukung dan mendoakan penulis.
9. Sahabat-sahabat penulis yang tidak bisa penulis sebutkan satu per satu, yang telah membantu penulis dan bertukar pikiran dalam penulisan proposal skripsi ini.

Akhirnya dengan segala kerendahan hati penulis mengharapkan skripsi ini dapat bermanfaat bagi pembaca dan kita semua yang membutuhkan.

Semarang, 01 Juli 2024

Penulis

Khoir Afandi

DAFTAR ISI

TUGAS AKHIR.....	i
MOTTO DAN PERSEMBAHAN.....	iii
PERNYATAAN KEASLIAN TULISAN	iv
ABSTRAK	v
PRAKATA.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Batasan Masalah.....	2
C. Identifikasi Masalah.....	2
D. Perumusan Masalah.....	3
E. Tujuan Penelitian.....	3
F. Manfaat Penelitian	3
BAB II KAJIAN PUSTAKA	5
A. Tinjauan Pustaka.....	5
B. Landasan Teori	9
C. Kerangka Berfikir	19
BAB III METODE PENELITIAN	22
A. Pendekatan Penelitian.....	22
B. Fokus Penelitian	22
C. Desain Penelitian	23
D. Teknik Pengumpulan Data	24

BAB IV HASIL DAN PEMBAHSAN.....	25
A. Hasil	25
1. Analisis Kebutuhan Sistem.....	25
2. Implementasi Model Pengenalan Huruf Bahasa Isyarat.....	26
3. Implementasi Sistem	34
B. Pembahasan	58
1. Analisis Kebutuhan Sistem.....	58
2. Implementasi Model Pengenalan Huruf Bahasa Isyarat.....	59
3. Implementasi Sistem	60
BAB V KESIMPULAN DAN SARAN	63
A. Kesimpulan	63
B. SARAN	64
DAFTAR PUSTAKA.....	65
LAMPIRAN.....	70
Lampiran 1 Lembar Bimbingan Dosen Pembimbing 1	70
Lampiran 2 Lembar Bimbingan Dosen Pembimbing 2	72
Lampiran 3 Kuisisioner Pengujian Black Box	73
Lampiran 4 Pengujian Use Acceptance Testing.....	79
Lampiran 5 Lembar Revisi Ujian	94
Lampiran 6 Dokumentasi UAT	97

DAFTAR TABEL

Tabel 2. 1 Tabel Tinjauan Pustaka.....	5
Tabel 2. 2 Use Case Diagram.....	16
Tabel 2. 3 Activity Diagram.....	17
Tabel 4. 1 augmentasi.....	27
Tabel 4. 2 Pelatihan Model.....	31
Tabel 4. 3 <i>Requirement</i>	43
Tabel 4. 4 <i>White Box Testing</i>	50
Tabel 4. 5 hasil Black Box Testing.....	54
Tabel 4. 6 hasil UAT.....	57

DAFTAR GAMBAR

Gambar 2. 1 Gambar CNN.....	10
Gambar 2. 2 Convolutional Layer.....	11
Gambar 2. 3 Pooling Layer	11
Gambar 2. 4 gambar prototype.....	15
Gambar 2. 5 kerangka berfikir	21
Gambar 3. 1 dataset.....	24
Gambar 4. 1 hasil layer yang ditambahkan.....	30
Gambar 4. 2 grafik hasil pelatihan model	32
Gambar 4. 3 hasil evaluasi model	33
Gambar 4. 4 hasil confusion matrix.	33
Gambar 4. 5 use case diagram.....	35
Gambar 4. 6 activity diagram upload	36
Gambar 4. 7 activity diagram realtime.....	37
Gambar 4. 8 Activity Diagram penerjemah	37
Gambar 4. 9 <i>Wireframe</i> Halaman home	38
Gambar 4. 10 <i>Wireframe</i> Halaman home (realtime).....	39
Gambar 4. 11 <i>protoype</i> halaman home	40
Gambar 4. 12 protoype halaman home realtime	41
Gambar 4. 13 Inialisasi Aplikasi Flas.....	44
Gambar 4. 14 Label Kelas.....	44
Gambar 4. 15 rute prediksi.....	45
Gambar 4. 16 deteksi upload gambar.....	46
Gambar 4. 17 deteksi realtime	47
Gambar 4. 18 hasil website halaman home.....	48
Gambar 4. 19 hasil website halaman home realtime.....	49
Gambar 4. 20 Flowgraph.....	52
Gambar 4. 21 hasil augmentasi	59
Gambar 4. 22 hasil trainig.....	60

DAFTAR LAMPIRAN

Lampiran 1 Lembar Bimbingan Dosen Pembimbing 1	70
Lampiran 2 Lembar Bimbingan Dosen Pembimbing 2	72
Lampiran 3 Kuisisioner Pengujian Black Box	73
Lampiran 4 Pengujian Use Acceptance Testing.....	79
Lampiran 5 Lembar Revisi Ujian.....	94
Lampiran 6 Dokumentasi UAT	97

BAB I

PENDAHULUAN

A. Latar Belakang

Komunikasi merupakan hal yang penting bagi manusia, baik untuk berinteraksi dengan sesama maupun untuk menyampaikan informasi. Namun, bagi orang tuna rungu dan tuna wicara, komunikasi dapat menjadi tantangan tersendiri. Hal ini dikarenakan mereka menggunakan bahasa isyarat sebagai alat komunikasi. Bahasa isyarat merupakan bahasa komunikasi non-verbal yang menggunakan gerakan tangan, gerak tubuh, dan ekspresi wajah untuk menyampaikan maksud tertentu. Bahasa isyarat memiliki kosakata dan tata bahasa yang berbeda-beda di setiap negara. Di Indonesia, bahasa isyarat yang umum digunakan adalah Bahasa Isyarat Indonesia (BISINDO) [1]. Menurut jurnal "Bahasa Isyarat Indonesia" yang ditulis oleh Danti Ayu Saraswati, Vera Diana Towidjojo, dan Hasanuddin dalam Jurnal Medical Profession (MedPro), Vol. 4 No. 1 Februari 2022, hasil kuesioner dan wawancara terhadap 100 responden tuna rungu usia remaja dan dewasa (16-50 tahun) di berbagai daerah, bahwa mayoritas tunarungu, yaitu 91%, telah memakai BISINDO dalam komunikasi antar sesama manusia, sedangkan SIBI hanya dipakai oleh 9%.

Orang tuna rungu dan tuna wicara sering kali mengalami kesulitan berkomunikasi dengan orang normal yang tidak menguasai bahasa isyarat. Hal ini dapat menimbulkan hambatan dalam berbagai aspek kehidupan, seperti pendidikan, pekerjaan, dan interaksi sosial. Untuk mengatasi masalah tersebut, dibutuhkan alat bantu yang dapat menjembatani komunikasi antara orang tuna rungu dan orang normal. Salah satu alat bantu yang dapat digunakan adalah aplikasi pengenalan Bahasa isyarat.

Aplikasi pengenalan huruf abjad bahasa isyarat menggunakan metode *Convolutional Neural Network* (CNN) adalah sebuah pengenalan dan juga penerjemah bahasa isyarat secara otomatis. Sistem ini berbeda dengan metode manual yang melibatkan penerjemah bahasa isyarat manusia. Dalam aplikasi ini, proses penerjemahan dilakukan secara otomatis oleh komputer, tanpa keterlibatan penerjemah manusia. Dikembangkan khusus untuk mempermudah

pembelajaran bahasa isyarat, aplikasi ini ditujukan bagi penyandang tuna rungu, tuna wicara, dan masyarakat umum yang ingin mempelajari bahasa isyarat dengan bantuan teknologi terkini. Dengan menggunakan metode CNN, aplikasi ini diharapkan dapat mengenali dan memahami gerakan tangan yang merupakan huruf abjad dalam bahasa isyarat dengan baik, memberikan pengalaman pembelajaran yang efisien dan mudah diakses bagi pengguna. Salah satu metode yang dapat digunakan untuk membangun sistem penerjemah bahasa isyarat secara otomatis adalah metode CNN, CNN merupakan salah satu metode *deep learning* yang mampu mengenali pola pada citra [2].

Penelitian ini bertujuan untuk mengembangkan dan menguji keefektifan dalam mengenali bahasa isyarat menggunakan metode CNN. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan teknologi untuk meningkatkan aksesibilitas komunikasi bagi penyandang tuna rungu dan tuna wicara dapat berinteraksi lebih baik antara mereka dan masyarakat umum [3].

B. Batasan Masalah

Berdasarkan latar belakang diatas, batasan masalah yang diangkat berkaitan dengan penelitian yang dilakukan sebagai berikut:

1. Sistem ini hanya terfokus pada terjemahan Bahasa Isyarat Indonesia (BISINDO).
2. Membuat sistem penerjemah Bahasa isyarat yg menerjemah Bahasa isyarat untuk huruf A-Z.
3. Akurasi pengenalan gestur oleh CNN tidak dapat mencapai 100%. Faktor-faktor seperti pencahayaan, kecepatan gerakan, dan variasi gestur dapat mempengaruhi akurasi pengenalan.

C. Identifikasi Masalah

Dari beberapa uraian yang dikemukakan pada latar belakang, maka dapat didefinisikan masalah-masalah sebagai berikut:

1. Kesulitan Komunikasi bagi Penyandang Tuna Rungu dan Tuna Wicara.
2. Keterbatasan Sistem Penerjemah Bahasa Isyarat.
3. Kurangnya Akses terhadap Pembelajaran Bahasa Isyarat.

D. Perumusan Masalah

Dari penjelasan dalam latar belakang diatas maka dapat di rumuskan permasalahannya adalah “Bagaimana membangun aplikasi pengenalan huruf abjad bahasa isyarat menggunakan metode Convolutional Neural Network (CNN) berbasis React js”

E. Tujuan Penelitian

Membangun aplikasi pengenalan huruf bahasa isyarat menggunakan metode CNN berbasis *react.js* yang akurat dan dapat digunakan dalam situasi sehari-hari.

F. Manfaat Penelitian

Penelitian ini diharapkan memberikan sejumlah manfaat. Beberapa manfaat yang diharapkan dari penelitian ini antara lain:

1. Bagi Penulis

Menerapkan pengetahuan dan keterampilan yang saya dapatkan selama studi di program Informatika Universitas PGRI Semarang dalam kehidupan sehari-hari membawa manfaat besar bagi saya. Dengan menggunakan teknologi yang saya pelajari, saya bisa mengembangkan solusi sederhana untuk masalah di sekitar saya. Misalnya, saya bisa membuat aplikasi sederhana untuk membantu manajemen tugas sehari-hari, atau sistem untuk mencatat inventaris di rumah atau di tempat kerja.

2. Bagi pengguna

- a. Aksesibilitas Komunikasi yang Lebih Baik

Aplikasi pengenalan bahasa isyarat yang dikembangkan dapat memberikan penyandang tuna rungu dan tuan wicara aksesibilitas komunikasi yang lebih baik dengan masyarakat umum. Ini membantu mengurangi hambatan komunikasi yang sering dihadapi oleh mereka.

- b. Peningkatan Interaksi Sosial

Dengan adanya aplikasi pengenalan bahasa isyarat yang efektif, diharapkan akan terjadi peningkatan interaksi sosial antara penyandang tuna rungu dan tuna wicara dengan non-tuna rungu. Hal ini dapat memberikan peluang untuk keterlibatan sosial yang lebih aktif.

3. Bagi Akademik

Sebagai bahan kajian atau sumber referensi untuk penelitian selanjutnya dan sebagai bahan acuan adek tingkat untuk melakukan penelitian.

BAB II KAJIAN PUSTAKA

A. Tinjauan Pustaka

Tinjauan pustaka yang dipergunakan dalam penelitian ini disajikan melalui analisa perbandingan. Peneliti terdahulu juga bertujuan untuk memberikan kontribusi peneliti bagi penulis agar peneliti dengan tema ini dapat terus berkembang. Namun setiap penelitian memiliki pola dan kriteria yang berbeda satu dengan lainnya. Berikut merupakan analisa perbandingan penelitian terdahulu:

Tabel 2. 1 Tabel Tinjauan Pustaka

No	Penulis	Judul Penelitian	Metode	Hasil
1.	Mochamad Bagus Setyo Bakti (2019)	Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Metode Convolution Neural Network	Convolution Neural Network (CNN)	Penelitian ini menggunakan metode CNN untuk Sistem Isyarat Bahasa Indonesia (CNN) mendeteksi angka Sistem Isyarat Bahasa Indonesia (SIBI). Penelitian ini menghasilkan akurasi mencapai 98.89%. Dan mengalami kegagalan pada proses deteksi angka 2

2.	Indra Jiwana Thira (2023)	Pengenalan Alfabet Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Convolutional Neural Network	Convolution Neural Network (CNN)	Hasil penelitian menunjukkan bahwa dalam perbandingan tiga arsitektur Convolutional Neural Network (CNN) MobileNet, yaitu MobileNetV2, MobileNetV3Small, dan MobileNetV3Large, MobileNetV3Small adalah yang paling optimal. Model yang dihasilkan oleh MobileNetV3Small mencapai akurasi sebesar 98,81% saat diuji menggunakan data tes.
3.	Joelyan Vicky (2023)	Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN	Convolution Neural Network (CNN)	Dari penelitian yang dilakukan dapat ditarik kesimpulan yaitu berdasarkan proses implementasi yang dilakukan dengan model dan testing data yang dijalankan pada aplikasi berbasis mobile application ini. Terhadap data daun alpukat yang terkena penyakit mempunyai hasil yang baik dimana metode CNN mampu mendeteksi penyakit pada daun alpukat dengan tingkat akurasi sebesar 80%

4.	Zahrah Fadhilah (2023)	Pengenalan Alfabet SIBI Menggunakan Convolutional Neural Network sebagai Media Pembelajaran Bagi Masyarakat Umum	Convolution Neural Network (CNN)	Penelitian menunjukkan bahwa model Convolutional Neural Network (CNN) efektif dalam mengenali citra Alfabet SIBI. Percobaan terbaik dilakukan pada pelatihan ke-10 dengan 50 epoch dan batch size 8, mencapai akurasi validasi 87% dan pelatihan 85%. Pengujian model berhasil mengenali 457 dari 520 data testing.
5.	Reza Haris Alfikri (2022)	Pembangunan aplikasi penerjemah bahasa isyarat dengan metode CNN (Convolutional Neural Network) berbasis Android	Convolutional Neural Network (CNN)	Penelitian ini berhasil mengembangkan model guna untuk melakukan prediksi images dari bahasa isyarat dengan menggunakan metode Convolutional Neural Network (CNN). Pengembangan dilakukan dengan menggunakan Pre-Trained Model dengan tujuan untuk mendapatkan hasil akurasi yang tinggi. Pre-Trained Model yang digunakan adalah berjenis Efficient-Net Lite1 dengan hasil akurasi yang didapatkan mencapai 97% dengan 200 Epoch.

Penelitian mengenai “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network” yang dilakukan oleh Mochamad Bagus Setiyo Bakti (2019). Dalam penelitian ini menggunakan metode *Convolutional Neural Network* dalam mengenali angka Sistem Isyarat Bahasa Indonesia. Pada penelitian ini proses training menghasilkan akurasi berbeda dari setiap tahapnya yaitu dengan 24 epoch sampai 100 epoch dapat mencapai tingkat akurasi 96.44% rcobaan yang dilakukan, proses prediksi angka dua [4].

Perbandingan dari masing-masing Model berhasil dilakukan dan mendapatkan model yang dihasilkan *MobileNetV3Small* dengan jumlah epoch 30 menjadi model yang paling baik diantara model lain yang dihasilkan selama penelitian dan memiliki akurasi training sebesar 100% dan validation sebesar 97,62%. Pengujian model terbaik terhadap 84 data testing mendapatkan akurasi sebesar 98,81% dengan prediksi benar pada 84 citra dan prediksi salah pada satu citra. Hasil evaluasi model terbaik dalam melakukan klasifikasi pada data testing menghasilkan nilai recall 0,99, nilai presisi 0,99 dan nilai f1-score sebesar 0,99. Penelitian selanjutnya bisa menggunakan dataset beragam terutama dari sisi latar belakang pengambilan citra agar mendapatkan model yang lebih baik lagi dalam melakukan klasifikasi pada berbagai jenis citra alfabet SIBI [5].

Dari penelitian yang dilakukan dapat ditarik kesimpulan yaitu berdasarkan proses implementasi yang dilakukan dengan model dan testing data yang dijalankan pada aplikasi berbasis mobile application ini. Terhadap data daun alpukat yang terkena penyakit mempunyai hasil yang baik dimana metode CNN mampu mendeteksi penyakit pada daun alpukat dengan tingkat akurasi sebesar 80% untuk pencocokan jenis penyakit daun alpukat yang teridentifikasi pada database yang tersimpan di sistem mobile tersebut [6].

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa metode *Convolutional Neural Network* (CNN) berhasil mengenali citra Alfabet SIBI dengan efektif. Model yang dihasilkan dari pelatihan menggunakan 2.600 citra, dengan struktur arsitektur yang melibatkan tiga *Convolution Layer*, tiga

Pooling Layer, dan *Fully Connected Layer*. Percobaan pelatihan ke-10 menunjukkan hasil terbaik, dimana model tersebut menggunakan 50 epoch dan batch size 8, dengan akurasi validasi mencapai 87% dan akurasi pelatihan mencapai 85%. Hal ini menunjukkan bahwa penyesuaian yang tepat pada batch size dan jumlah epoch dapat menghasilkan kinerja model yang optimal. Selain itu, pada proses pengujian, model mampu mengenali sebanyak 457 data dari total 520 data testing, dengan akurasi rata-rata sebesar 87.9%. Hasil ini menegaskan bahwa model yang dihasilkan dari pelatihan mampu menggeneralisasi dengan baik pada data uji yang tidak digunakan sebelumnya, dan dapat diandalkan dalam pengenalan citra Alfabet SIBI [7].

Penelitian ini berhasil mengembangkan model guna untuk melakukan prediksi images dari bahasa isyarat dengan menggunakan metode *Convolutional Neural Network* (CNN). Pengembangan dilakukan dengan menggunakan *Pre-Trained Model* dengan tujuan untuk mendapatkan hasil akurasi yang tinggi. *Pre-Trained Model* yang digunakan adalah berjenis *Efficient-Net Lite1* dengan hasil akurasi yang didapatkan mencapai 97% dengan 200 Epoch [8].

B. Landasan Teori

Landasan teori merupakan penjelasan dari konsep dasar dan teori-teori yang berkaitan dengan penelitian. Untuk mendukung penyusunan secara penyusunan penelitian, dibutuhkan landasan teori yang mendukung penyusunan searah dengan landasan teori tersebut.

1. Bahasa isyarat

Bahasa isyarat merupakan Bahasa non-verbal, yang mengutamakan gerakan beserta ekspresi yang digunakan oleh masyarakat yang disebut teman tuli. Bahasa ini memiliki tata bahasa dan strukturnya sendiri, berbeda dengan bahasa lisan. Bahasa isyarat menggunakan kombinasi gerakan tangan, ekspresi wajah, dan bahasa tubuh untuk menyampaikan makna [9].

2. Bahasa Isyarat Indonesia (BISINDO).

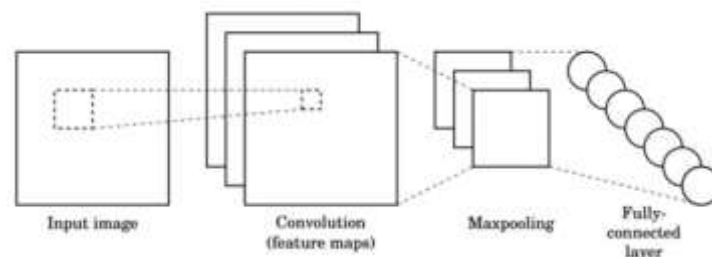
BISINDO adalah bahasa yang digalakkan oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN) dan dikembangkan oleh masyarakat tunarungu sendiri [10]. Bahasa ini merupakan hasil dari pengembangan dari

bahasa Indonesia yang merupakan bahasa ibu yang digunakan dalam kehidupan sehari-hari. Karena itu, BISINDO lebih umum digunakan oleh komunitas tuna rungu di Indonesia.

3. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah salah satu jaringan saraf tiruan *deep feedforward* yang luas diterapkan pada visi komputer. Metode ini juga dikenal sebagai *ConvNet*. Mirip dengan jaringan syaraf tiruan, CNN memiliki arsitektur yang terdiri dari *node* atau *neuron* yang saling terhubung [11]. *Neuron* yang saling terhubung satu sama lain dalam sebuah lapisan. CNN juga termasuk dalam kategori *Deep Neural Network* karena memiliki struktur yang dalam dan sering digunakan dalam pengolahan citra. Metode CNN terdiri dari dua tahap, yaitu klasifikasi yang menggunakan *feedforward* dan proses pembelajaran yang menggunakan algoritma *backpropagation*.

Selanjutnya dalam Metode *Convolutional Neural Network (CNN)* memiliki sebuah arsitektur. Arsitektur CNN adalah sebuah jaringan saraf yang menggunakan sebuah layar *konvolusi* dan *max pooling*. Seperti gambar dibawah ini



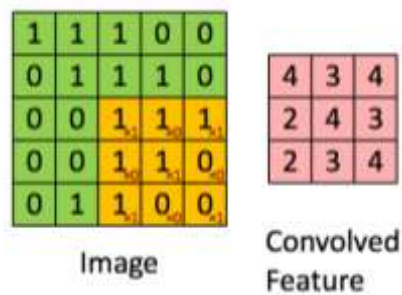
Gambar 2. 1 Gambar CNN

CNN umumnya terdiri dari beberapa lapisan, yang dapat dikelompokkan menjadi tiga jenis utama:

a. Lapisan Konvolusi (*Convolutional Layer*)

Convolutional layer merupakan lapisan pertama pada struktur CNN yang merupakan inti dari CNN dan tempat terjadinya sebagian besar proses komputasi. Pada lapisan ini dilakukan proses konvolusi dengan menggeser matriks filter ke seluruh permukaan citra [12]. Lapisan ini

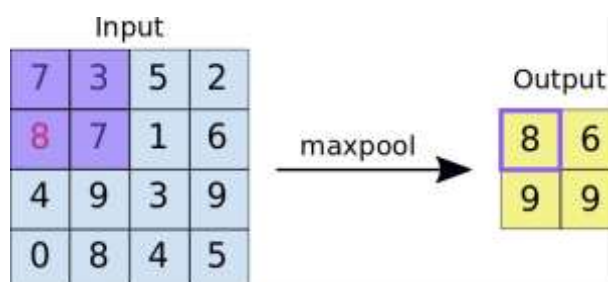
adalah inti dari arsitektur CNN. Fungsi utamanya adalah untuk mengekstraksi fitur dari input data menggunakan filter (atau *kernel*) yang bergerak melintasi input. Setiap filter mendeteksi berbagai pola seperti tepi, tekstur, atau objek sederhana dalam gambar. Hasilnya adalah peta fitur (*feature map*) yang menunjukkan lokasi dan kekuatan fitur yang terdeteksi.



Gambar 2. 2 Convolutional Layer

b. Lapisan Pooling (*Pooling Layer*)

Setelah konvolusi, lapisan pooling digunakan untuk mengurangi dimensi spasial (panjang dan lebar) dari peta fitur sambil tetap mempertahankan informasi penting. *Pooling* bertujuan untuk memperkecil ukuran matrik agar perhitungan yang dilakukan mesin menjadi lebih cepat [13]. Ini dilakukan dengan cara mengambil nilai maksimum (*max pooling*) atau rata-rata (*average pooling*) dari patch kecil dalam peta fitur. Pooling membantu mengurangi jumlah parameter, mencegah *overfitting*, dan membuat jaringan lebih tahan terhadap perubahan kecil pada input.



Gambar 2. 3 Pooling Layer

c. Lapisan Terhubung Sepenuhnya (*Fully Connected Layer*)

Fully connected layer adalah lapisan dimana semua neuron aktif pada lapisan sebelumnya terhubung dengan neuron pada lapisan berikutnya, mirip dengan jaringan syaraf tiruan [14]. Di tahap akhir, output dari lapisan konvolusi dan pooling dilewatkan ke satu atau lebih lapisan terhubung sepenuhnya (*fully connected layer*). Setiap neuron di lapisan ini terhubung ke semua neuron di lapisan sebelumnya, mirip dengan jaringan saraf tradisional. Lapisan ini bertanggung jawab untuk menggabungkan fitur-fitur yang telah diekstraksi untuk membuat keputusan akhir, seperti klasifikasi objek dalam gambar.

4. *Deep learning*

Deep learning adalah bagian dari pembelajaran mesin yang berkaitan dengan algoritma dimana cara kerja dari algoritma ini meniru struktur dan fungsi otak yang disebut jaringan saraf tiruan [15]. Penelitian terkait *deep learning* telah banyak dilakukan dalam berbagai bidang. Dalam *deep learning*, metode CNN sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *deep learning* penggunaan GPU sudah sangatlah umum.

5. *Transfer learning*

Transfer learning adalah suatu metode pembelajaran yang mendalam di mana pengetahuan yang diperoleh dari mempelajari satu masalah digunakan untuk memecahkan masalah yang berbeda-beda [16]. Pendekatan ini secara efisien membantu dalam memanfaatkan pengetahuan yang sudah ada untuk meningkatkan performa model pada tugas yang spesifik tanpa harus melakukan pelatihan model dari awal. Hal ini sangat berguna ketika dataset pelatihan berjumlah terbatas serta dapat menghemat waktu dan sumber daya komputasi yang digunakan selama pelatihan model yang kompleks.

MobileNetV2 merupakan model yang diusulkan oleh *Google* pada tahun 2017. Arsitektur ini dirilis pada awal tahun 2018 sebagai pengembangan dari model sebelumnya yaitu *MobileNetV1* yang diusulkan pada tahun 2015. Pengembangan dari model ini terlihat pada munculnya bottleneck linear di antara layer dan koneksi cepat antar *bottleneck*. Bottleneck pada *MobileNetV2* membantu mengemas model dari konsep tingkat rendah menjadi deskriptor tingkat tinggi sehingga model dapat dilatih dengan lebih cepat dan menghasilkan akurasi tinggi meskipun model menggunakan parameter yang lebih sedikit [17].

6. *Website*

Website merupakan kumpulan informasi yang terdiri dari halaman web yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi [18]. *Website* bersifat unik karena setiap halaman dapat saling terhubung dengan menggunakan hipertaut (*hyperlink*), baik dalam domain yang sama maupun berbeda. Umumnya website memuat informasi yang disediakan secara perorangan, kelompok, atau organisasi.

7. *Flask*

Flask adalah sebuah *micro-framework* berbasis bahasa *Python* yang tidak memiliki banyak *tools* dan *library*. Untuk membuat pengembangan lebih efisien maka digunakan *Flask*. *Flask* diterapkan karena dapat berjalan pada program yang berkapasitas energi kecil serta dengan memory yang rendah sehingga tidak memerlukan *resource* yang besar. Walaupun *framework Flask* ringan akan tetapi fungsinya masih bisa berjalan sesuai yang dibutuhkan [19].

8. Metode *Prototype*

Metode *Prototype* merupakan metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna. Model *prototype* yang digunakan oleh sistem akan mengizinkan pengguna mengetahui seperti apa tahapan sistem yang dibuat sehingga sistem mampu beroperasi secara baik [20]. Metode *prototype* yang diterapkan pada penelitian ini dimaksudkan agar

mendapatkan representasi dari pemodelan aplikasi yang akan dibuat, berikut adalah tahapan dalam metode *prototype*:

a. Pengumpulan Kebutuhan:

Pada tahapan ini, informasi dikumpulkan dari pengguna untuk memahami kebutuhan dan tujuan perangkat lunak yang akan dibangun.

b. Membangun Prototype:

Berdasarkan kebutuhan yang telah dikumpulkan, pengembang membuat prototipe awal. Prototipe ini adalah representasi sederhana dari sistem yang diinginkan dan berfokus pada fungsi-fungsi utama. Tujuan utama dari prototipe ini adalah untuk menunjukkan konsep dan mendapatkan umpan balik awal dari pengguna.

c. Evaluasi Prototype:

Prototype yang telah dibuat dievaluasi oleh pengguna, untuk menguji dan memberikan umpan balik mengenai fitur, antarmuka, dan seluruh pengalaman penggunaan prototipe tersebut

d. Mengkodekan Sistem:

Setelah mendapatkan umpan balik dari evaluasi prototipe, pengembang mulai menulis kode untuk sistem yang sebenarnya.

e. Menguji Sistem:

Sistem yang telah dikodekan diuji untuk memastikan bahwa semua fungsionalitas bekerja sebagaimana mestinya dan tidak ada bug atau kesalahan.

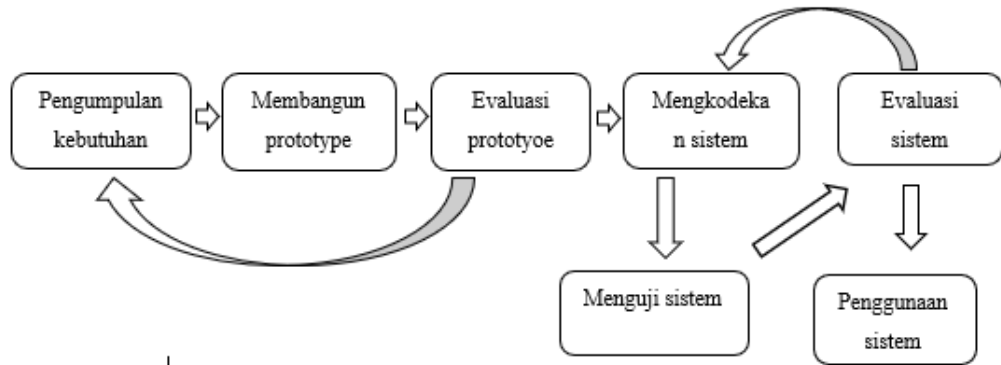
f. Evaluasi Sistem:

Setelah pengujian, sistem dievaluasi secara keseluruhan untuk memastikan bahwa memenuhi kebutuhan pengguna dan bekerja sesuai dengan spesifikasi yang telah ditetapkan.

g. Penggunaan Sistem:

Setelah sistem lolos evaluasi, sistem siap digunakan oleh pengguna akhir. Tahap ini mencakup pelatihan pengguna,

instalasi, dan dukungan teknis awal untuk memastikan bahwa pengguna dapat menggunakan sistem dengan efektif.



Gambar 2. 4 gambar metode prototype

9. UML



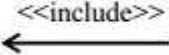
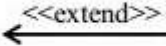




Unified Modeling Language (UML) merupakan satu set standar teknik diagram yang memberikan representasi grafis yang cukup kaya untuk model setiap pengembangan sistem proyek dari analisis melalui implementasi [21]. UML menyediakan notasi grafis untuk menggambarkan berbagai aspek sistem, seperti struktur, perilaku, dan interaksi antar komponen.

Beberapa komponen yang terdapat pada UML adalah sebagai berikut:

a. *Use Case Diagram*

Use case diagram merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna system. *Use case diagram* menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk actor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram *UML use case* [22].

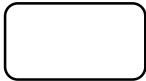




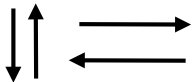
Tabel 2. 2 Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Mewakili peran orang, system yang lain ,atau alat berkomunikasi dengan <i>use case</i>
2		<i>Generalization</i>	Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i> .
3		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya suatu fungsionalitas dari <i>use case</i> lainnya
4		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi
5		<i>Association</i>	Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor.
8		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity diagram*

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu *system* [23]. *Diagram* ini menunjukkan serangkaian aktivitas atau tindakan yang terjadi dalam suatu proses, serta urutan dan kondisi di mana aktivitas tersebut dilakukan

Tabel 2. 3 Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
5		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
6		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

10. *React Js*

React Js adalah *library* untuk membuat *user interface* pada *website*. *React* diciptakan oleh *Meta* (dahulu *Facebook*) dan mulai dipublikasi pada *NPM repository* sejak tahun 2012. Dengan fitur seperti *Virtual DOM* dan konsep komponen, *React.js* memungkinkan pembangunan aplikasi *web* yang lebih mudah dipelajari dan dikembangkan [24].

11. *White Box Testing*

Pengujian *White Box*, atau pengujian kotak kaca, adalah metode pengujian perangkat lunak yang berfokus pada struktur internal dan kode program [26]. Pengujian ini berfokus pada bagaimana program tersebut bekerja secara internal, dengan menganalisis kode program dan melacak aliran data dan control, Karakteristik dari penggunaan *white box* adalah sebagai berikut:

- a) Fokus pada struktur internal: Pengujian ini memastikan apakah kode program ditulis dengan benar dan sesuai dengan spesifikasi.
- b) Membutuhkan pengetahuan kode: Penguji perlu memiliki pengetahuan tentang kode program untuk melakukan pengujian *white box*.
- c) Dapat menemukan error yang tidak terdeteksi oleh pengujian *black box*: Pengujian ini dapat menemukan error yang terkait dengan struktur internal program, seperti *loop* tak terbatas dan kondisi yang tidak terduga.
- d) Lebih kompleks dan memakan waktu: Pengujian ini lebih kompleks dan memakan waktu dibandingkan dengan pengujian *black box*.

12. Pengujian *Black Box*

Metode *black box* adalah sebuah pendekatan dalam pengujian perangkat lunak di mana pengujian dilakukan tanpa harus memperhatikan rincian internal dari program yang akan diuji. Dalam metode ini, fokus utama adalah pada nilai-nilai masukan yang diberikan kepada program tanpa perlu memperhatikan bagaimana program tersebut menghasilkan keluaran atau output [25].

Keuntungan utama dari penggunaan metode *black box* adalah pengujian dapat dilakukan tanpa memerlukan pengetahuan mendalam tentang bagaimana program tersebut bekerja di dalamnya. Pendekatan ini melibatkan pengujian dari sudut pandang pengguna akhir, sehingga programmer dan pengujian saling bergantung satu sama lain. Metode *black box* bukanlah alternatif untuk teknik-program sistem informasi yang diuji, melainkan fokus pada spesifikasi fungsional perangkat lunak. Salah satu teknik yang digunakan dalam metode *black box* adalah teknik partisi setara, di mana nilai-nilai masukan dipartisi menjadi kelompok yang setara untuk pengujian.

13. UAT(*User Acceptance Testing*)

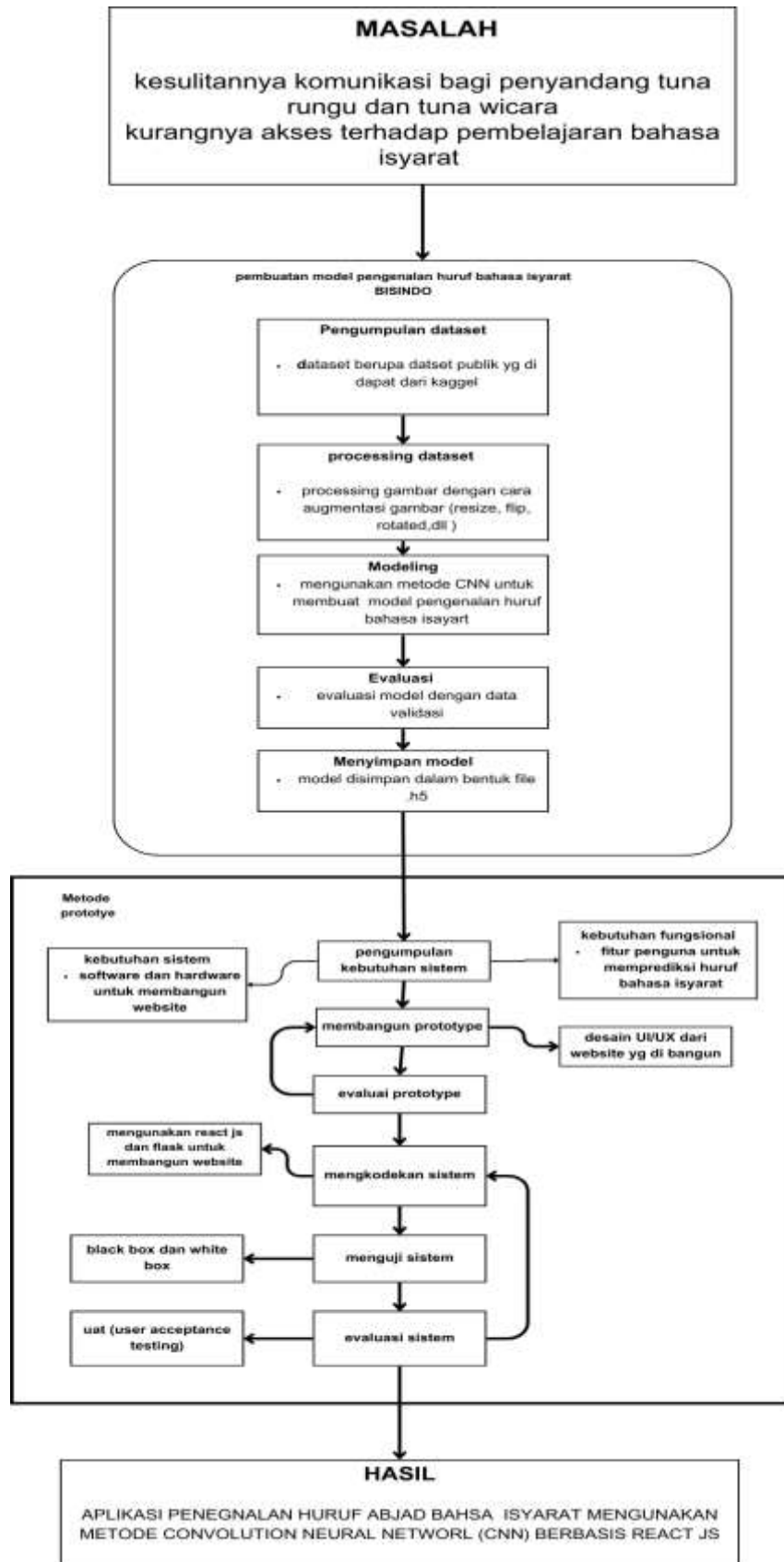
User Acceptance Testing merupakan pengujian yang dilakukan oleh end user yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya. UAT menguji yang dilakukan oleh pengguna sistem. Hasil dari pengujian dapat dijadikan bukti bahwa sistem dapat membantu para pengguna. UAT dilakukan pada pengembangan perangkat lunak bertujuan untuk memastikan 19 sistem memenuhi kebutuhan sebenarnya dari pengguna, bukan hanya spesifikasi 19 sistem [27].

Pengujian penerimaan pengguna (UAT) adalah fase terakhir dari proses pengujian perangkat lunak. Selama UAT, perangkat lunak perangkat lunak diuji untuk memastikan tugas-tugas apakah sudah sesuai dengan spesifikasinya. UAT adalah salah satu prosedur proyek perangkat lunak final dan paling penting yang harus terjadi sebelum perangkat lunak tersebut dikembangkan dan diluncurkan ke pasar. UAT juga dikenal sebagai pengujian beta, pengujian aplikasi atau pengujian pengguna akhir.

C. Kerangka Berfikir

Kerangka berpikir atau kerangka pemikiran adalah dasar pemikiran dari penelitian yang disintesis dari fakta-fakta, observasi dan kajian kepustakaan [28]. Kerangka berpikir ini dirancang untuk mempermudah proses penelitian dengan fokus pada pemecahan masalah komunikasi antara masyarakat umum dan teman penyandang tuli dan tuna wicara. Bahasa yang digunakan adalah

bahasa isyarat, yang tidak semua orang memahaminya. Dari permasalahan ini, penulis bertujuan untuk membangun sebuah aplikasi yang dapat menerjemahkan huruf bahasa isyarat Indonesia (BISINDO) ke dalam sebuah aplikasi Android menggunakan *algoritma Convolutional Neural Network (CNN)* dengan harapan dapat mendeteksi dengan keakuratan yang tinggi. Kerangka berpikir ini dapat dijelaskan sebagai berikut:



Gambar 2. 5 kerangka berfikir

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Pendekatan penelitian adalah serangkaian metode dan strategi yang dirancang untuk mengarahkan proses penelitian. Pendekatan ini melibatkan langkah-langkah yang didasarkan pada asumsi-asumsi tertentu untuk menentukan cara pengumpulan dan analisis data. Pada penelitian ini menggunakan metode *prototype*, Metode *Prototype* merupakan metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna.

Prototype digunakan untuk memberikan gambaran cepat tentang rancangan sistem informasi kepada pengguna tanpa memerlukan pengembangan penuh (*coding*). Tujuannya adalah agar antarmuka sistem informasi mudah dipahami oleh pengguna, terutama saat pengguna pertama kali menggunakannya. Metode ini melibatkan pengujian oleh pengguna untuk mendapatkan *evaluasi* yang kemudian digunakan sebagai dasar untuk pengembangan lebih lanjut. Dengan demikian, model *prototype* mengurangi waktu pengembangan serta menghindari kesalahan yang mungkin terjadi di tengah proses pengembangan sistem informasi[8].

B. Fokus Penelitian

Fokus penelitian difokuskan pada pembahasan sebagai berikut:

1. Aplikasi berjalan dengan baik
2. Aplikasi yg dihasilkan berbasis *website*
3. Aplikasi dapat mendeteksi huruf bahasa isyarat (BISINDO)
4. Target pengguna adalah orang penyandang tuna rungu dan wicara serta masyarakat umum yang ingin belajar bahasa isyarat.

C. Desain Penelitian

Desain penelitian merupakan langkah awal yang sangat penting dalam proses penelitian karena memberikan landasan yang jelas bagi peneliti dalam mengarahkan seluruh proses penelitian. Dalam melakukan penelitian terdapat beberapa tahapan dan skema sebagai berikut :

1. Tahap pembuatan model BISINDO

a. Pengumpulan dataset

Dataset berupa dataset publik huruf abjad Bahasa isyarat BISINDO yang didapat dari kaggle.

b. Processing dataset

Pada tahapan ini peneliti melakukan *preprocessing* dataset yg berupa gambar dengan melakukan augmentasi.

c. Modeling

Pemilihan metode CNN untuk membuat model pengenalan huruf Bahasa isyarat BISINDO.

d. Evaluasi

Evaluasi model dengan data validasi untuk mengevaluasi kinerja model.

e. Menyimpan model

Model disimpan dalam bentuk .h5 dan nantinya akan digunakan dalam pembuatan *website*.

2. Tahap *deploy* model dan pembuatan website

Pada tahapan ini penelitian ini menggunakan metode *prototype* untuk pembuatan *website*.

a. Pengumpulan kebutuhan sistem

kebutuhan sistem yg berupa *software* dan *hardware* untuk proses pembuatan *website*.

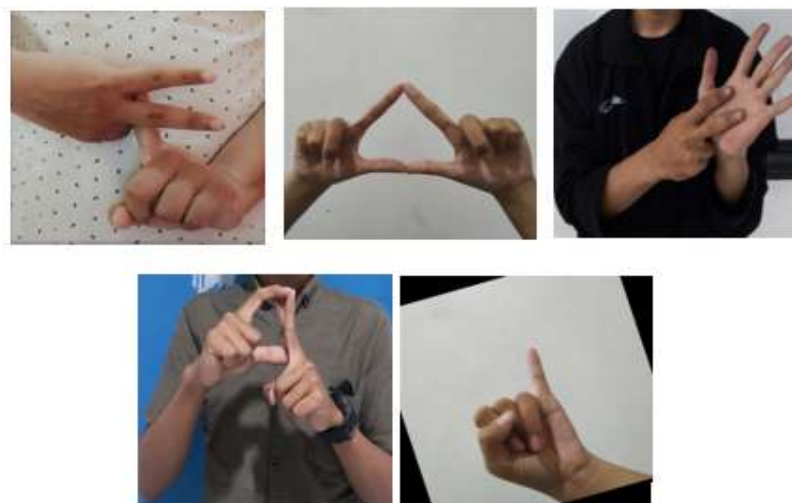
b. Membangun *prototype*

Prototype awal yaitu berupa desain UI/UX dari websute yg akan dibangun.

- c. Evaluasi *prototype*
Uji prototipe awal dengan sejumlah kecil pengguna untuk mendapatkan umpan balik awal.
- d. Mengkodekan sistem
Proses pembuatan *website* dan juga deploy model menggunakan *react js* dan *flask*.
- e. Menguji sistem
Proses pengujian *website* menggunakan pengujian *Black Box*, *White box*, Pengujian arsitektur, Basis path dan lain-lain.
- f. Evaluasi system
Pada tahap pengevaluasian ini dilakukan oleh klien untuk memastikan apakah program atau sistem yang sudah dibangun telah sesuai dengan keinginan atau belum.

D. Teknik Pengumpulan Data

Pada tahapan penelitian ini peneliti melakukan pengumpulan data yang berkaitan dengan subjek yang akan diangkat. Data yang dikumpulkan yaitu dataset publik dari kaggle yang berupa foto peragaan huruf alphabet dari Bahasa isyarat BISINDO mulai dari huruf A – Z yang berupa 11 huruf dalam bentuk tangan tunggal dan 15 bentuk tangan ganda, contoh dataset dapat dilihat pada gambar 3.1.



Gambar 3. 1 dataset

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas mengenai hasil dan pembahasan dari penelitian yang telah dilakukan oleh penulis yaitu aplikasi pengenalan huruf Bahasa isyarat

A. Hasil

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah langkah pertama dalam pengembangan suatu sistem atau proyek.

a. Kebutuhan Perangkat Keras:

Berikut ini adalah spesifikasi perangkat keras minimum yang mendukung untuk pembuatan aplikasi pengenlana huruf bahas isyarat:

1. Perangkat Keras Aplikasi Backend:

Processor : AMD E2 7th
RAM : 4GB
SSD : 256 GB
VGA : Minimum Integrated

2. Perangkat Keras Aplikasi Frontend:

OS : Minimum Windows 64 bit
RAM : Minimum 4GB
Hardisk : Minimum 4GB yang tersedia
Resolusi : 1280 x 800

b. Kebutuhan Perangkat lunak:

Berikut ini adalah spesifikasi perangkat lunak minimum yang akan digunakan untuk pembuatan aplikasi pengenlana huruf bahasa isyarat :

1. Google Colaboratory
2. Visual Studio Code
3. Postman
4. Figma
5. Kaggle
6. Google Drive

2. Implementasi Model Pengenalan Huruf Bahasa Isyarat

a. Pengumpulan Dataset

Data yang digunakan untuk membuat model adalah dataset yang berupa gambar huruf Bahasa isyarat dari A-Z di peroleh dari *website* Kaggle <https://www.kaggle.com/datasets/agungmrf/indonesian-sign-language-bisindo> . Dataset diproses untuk di dibagi menjadi dua folder yaitu *training data folder* dan *validation data folder*. Pembagian dataset yaitu 80:20 80% data untuk training dan 20% data untuk validasi.

1. *training data folder*

Training data folder digunakan untuk menyimpan dataset yang nantinya akan digunakan untuk melatih model. Di *Training data* ini terdapat 26 *class* yaitu *class* huruf A – Z dimana setiap *class* terdiri dari 300 gambar.

2. *validation datafolder*

validation data folder digunakan untuk menyimpan data yang digunakan untuk mengevaluasi kinerja model saat proses pelatihan. Di *validation data* ini terdapat 26 *class* dimana setiap *class* terdiri dari 75 gambar.

b. Pemrosesan Dataset

Dataset yang sudah ada akan dilakukan pemrosesan dataset dengan melakukan *agumentation* atau augmentasi. Menurut jurnal "Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data" yang diterbitkan di Algoritma: Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumian dan Angkasa, Vol. 2, No. 3 Mei 2024, pemrosesan dataset menggunakan teknik augmentasi gambar seperti rotation range, width shift range, height shift range, shear range, zoom range, dan horizontal flip itu penting. Augmentasi ini bertujuan meningkatkan kuantitas dan variasi data pelatihan, meningkatkan generalisasi model, menghindari overfitting, dan memperluas keragaman dataset [29]. Studi ini dilakukan oleh Mochammad Toyib, Tegar Decky Kurniawan Pratama, dan Ibnu Aqil dari Universitas Nurul Jadid. Artikel ini diterima pada tanggal 27 Mei

2024 dan dipublikasikan pada tanggal 31 Mei 2024. Dalam penelitian ini, penulis menggunakan berbagai teknik augmentasi data, seperti *Rotation Range*, *Width Shift Range*, *Height Shift Range*, *Horizontal Flip*.

Tabel 4. 1 *augmentasi*

NO	Augmentasi	Nilai	Penjelasan
1	<i>Rotation Range</i>	10	Rotasi acak pada gambar hingga sudut maksimum 10 derajat. Membantu model mengenali objek meskipun mengalami rotasi dalam data nyata.
2	<i>Width Shift Range</i>	0.1	Pergeseran <i>horizontal</i> acak pada gambar hingga 10% dari lebar gambar. Membantu model menangani variasi posisi objek dalam gambar.
3	<i>Height Shift Range</i>	0.1	Pergeseran <i>vertikal</i> acak pada gambar hingga 10% dari tinggi gambar. Membantu model menangani variasi posisi objek dalam gambar.
4	<i>Horizontal Flip</i>	<i>True</i>	Pembalikan <i>horizontal</i> acak pada gambar (<i>mirroring</i>). Meningkatkan keragaman data dengan menyertakan versi gambar yang diputar secara <i>horizontal</i> .

c. *Modeling*

Proses pembuatan model BISINDO, tahapan ini menggunakan teknik *Transfer learning* dan menggunakan arsitektur *MobileNetV2*.

1. Inisialisasi Model Pra-pelatihan

pada arsitektur ini di *MobileNetV2* menerapkan beberapa parameter sebagai berikut:

- a. *Include_top = False* yang berarti tidak menyertakan lapisan penuh (*fully connected layer*) karena nanti akan ditambahkan lapisan klasifikasi kustom di atas model.

- b. *Weights = imagenet* Parameter ini menggunakan bobot (*weights*) yang sudah dilatih sebelumnya pada dataset *ImageNet*. Ini membuat model sudah memiliki pengetahuan awal tentang gambar.
- c. *Input_shape = (224, 224, 3)* Parameter ini menentukan bentuk input untuk model. Bentuk input yang diharapkan adalah gambar dengan ukuran 224x224 piksel dengan 3 saluran warna (RGB).

2. Pengembangan Model Akhir

Pada tahap ini, kita melakukan penyesuaian pada model pra-terlatih (*MobileNetV2*) dengan menambahkan lapisan-lapisan baru untuk membentuk model akhir yang siap digunakan untuk tugas klasifikasi spesifik. Library utama yang digunakan dalam kode ini adalah *TensorFlow* dengan *API Keras*. Kode memanfaatkan komponen-komponen dari *Keras* seperti *MobileNetV2* untuk memuat model yang telah dilatih sebelumnya. Proses ini mencakup penambahan lapisan dense dan dropout, serta pembentukan lapisan output. Berikut penjelasan detailnya:

a. *Global Average Pooling*

Langkah pertama setelah model pra-pelatihan adalah menggunakan lapisan *Global Average Pooling*. Lapisan ini mengubah keluaran 4D dari *MobileNetV2* (*batch_size, height, width, channels*) menjadi bentuk 2D (*batch_size, channels*). Tujuannya adalah Mereduksi dimensi keluaran sambil mempertahankan informasi penting dari fitur yang telah dipelajari oleh model pra-terlatih.

b. Penambahan Lapisan Dense dan Dropout

1. *Dense Layers*

Pada *dense layer* yang termasuk Lapisan *fully connected* dengan sejumlah neuron yang berbeda (1024, 512, 256). Aktivasi *ReLU* digunakan untuk memperkenalkan *non-*

linearitas dalam model. Nama lapisan diberikan untuk identifikasi (*dense_1*, *dense_2*, *dense_3*).

2. *Dropout Layers*

Tahap ini menggunakan 0,5 untuk dropout, dengan tujuan mengurangi *overfitting* dengan mengatur 50% *neuron* secara acak menjadi tidak aktif selama pelatihan.

3. *Output Layer*

Dalam tahap ini menambahkan lapisan output ke model yang sudah dibangun sebelumnya. Lapisan output bertanggung jawab untuk menghasilkan prediksi akhir dari model.

Pada lapisan ini terdapat 26 neuron digunakan dalam lapisan ini. Jumlah ini ditentukan berdasarkan jumlah kelas dalam tugas klasifikasi. Disini untuk mengklasifikasi huruf abjad bahasa isyarat yang memiliki 26 kelas (A hingga Z). disini juga mengunakan fungsi *Aktivasi Softmax*, Fungsi softmax mengubah keluaran mentah dari lapisan dense menjadi probabilitas. Softmax mengambil nilai output dari neuron-neuron di lapisan terakhir dan menormalkannya menjadi rentang antara 0 dan 1 sehingga jumlah seluruh probabilitas adalah 1. Berikut ringkasan layer yang sudah di tambahkan dapat dilihat pada Gambar 4.1

conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	['conv_1[0][0]']
out_relu (ReLU)	(None, 7, 7, 1280)	0	['conv_1_bn[0][0]']
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1280)	0	['out_relu[0][0]']
dense_1 (Dense)	(None, 1024)	1311744	['global_average_pooling2d_1[0][0]']
dropout_3 (Dropout)	(None, 1024)	0	['dense_1[0][0]']
dense_2 (Dense)	(None, 512)	524800	['dropout_3[0][0]']
dropout_4 (Dropout)	(None, 512)	0	['dense_2[0][0]']
dense_3 (Dense)	(None, 256)	131328	['dropout_4[0][0]']
dropout_5 (Dropout)	(None, 256)	0	['dense_3[0][0]']
output (Dense)	(None, 26)	6682	['dropout_5[0][0]']

Total params: 4232538 (16.15 MB)			
Trainable params: 4198426 (16.02 MB)			
Non-trainable params: 34112 (133.25 KB)			

Gambar 4. 1 hasil layer yang ditambahkan

d. Pelatihan Model

Setelah model akhir dibentuk dengan menambahkan lapisan kustom dan lapisan output, langkah berikutnya adalah proses pelatihan model

1. Kompilasi Model

kompilasi model adalah langkah pertama sebelum pelatihan dimulai. Dalam proses ini, menentukan optimasi yang akan digunakan, fungsi loss, dan metrik evaluasi.

- a. *Optimizer*: *adam* adalah algoritma optimasi adaptif yang populer dan efisien untuk pelatihan jaringan saraf. *Optimizer adam* dipilih karena mudah diterapkan, lebih efisien secara komputasi dan kebutuhan memori yang lebih kecil.
- b. *Loss Function*: *categorical_crossentropy* digunakan untuk tugas klasifikasi multi-kelas. Fungsi ini menghitung perbedaan antara distribusi prediksi dan distribusi target.
- c. *Metrics*: *accuracy* adalah metrik yang umum digunakan untuk menilai kinerja model klasifikasi, menunjukkan persentase prediksi yang benar.

2. Pelatihan Model

Proses pelatihan melibatkan penggunaan generator data untuk memasukkan data ke dalam model, menjalankan pelatihan selama beberapa *epoch* dan juga menggunakan teknik *callback*.

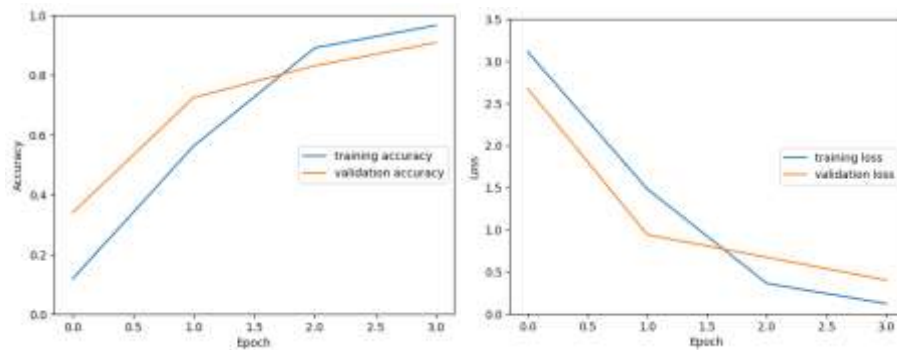
Library utama yang digunakan dalam pelatihan model ini adalah *TensorFlow* dengan *API Keras*. Dengan komponen *Callback*, khususnya *EarlyStopping*, digunakan untuk menghentikan pelatihan lebih awal jika diperlukan. *Optimizer Adam* digunakan untuk memperbarui bobot model, dan metode *model.fit* digunakan untuk melatih model dengan data yang disediakan oleh generator, serta mengelola proses pelatihan secara keseluruhan. Proses pelatihan model dimulai dengan menggunakan metode *fit* untuk melatih model dengan generator data pelatihan dan validasi. Metode *fit* ini mengambil beberapa parameter penting untuk mengontrol pelatihan, yaitu *steps_per_epoch*, *validation_steps*, dan *epochs*.

Dalam pembuatan model ini, model dilatih menggunakan *train_generator* sebagai sumber data pelatihan dan *validation_generator* sebagai sumber data validasi. *steps_per_epoch* dan *validation_steps* dihitung berdasarkan jumlah sampel dalam generator masing-masing, dan model dilatih selama 50 *epoch*. Proses ini memastikan bahwa model dilatih secara menyeluruh dengan menggunakan seluruh dataset beberapa kali, dan dievaluasi pada data validasi secara berkala untuk memonitor kinerjanya. Dalam pembuatan model ini juga menggunakan menggunakan *callback* jenis *early stop*, *callback* ini digunakan digunakan untuk menghentikan proses pelatiha jika akurasi model mencapai lebih dari 96%. Untuk rincian pada pelatihan model dapat dilihat pada Tabel 4 2.

Tabel 4. 2 Pelatihan Model

No	<i>Epoch</i>	<i>Accuracy</i>		<i>Loss</i>	
		<i>Acc</i>	<i>Val_acc</i>	<i>Loss</i>	<i>Val_loss</i>
1	<i>Epoch 1</i>	0.1173	0.3390	3.1110	2.6737
2	<i>Epoch 2</i>	0.5622	0.7241	1.4806	0.9377
3	<i>Epoch 3</i>	0.8906	0.8308	0.3584	0.6690

4	<i>Epoch 4</i>	accuracy: 0.9662 Accuracy -- Stop Training			
		0.9662	0.9082	0.1171	0.3965



Gambar 4. 2 grafik hasil pelatihan model

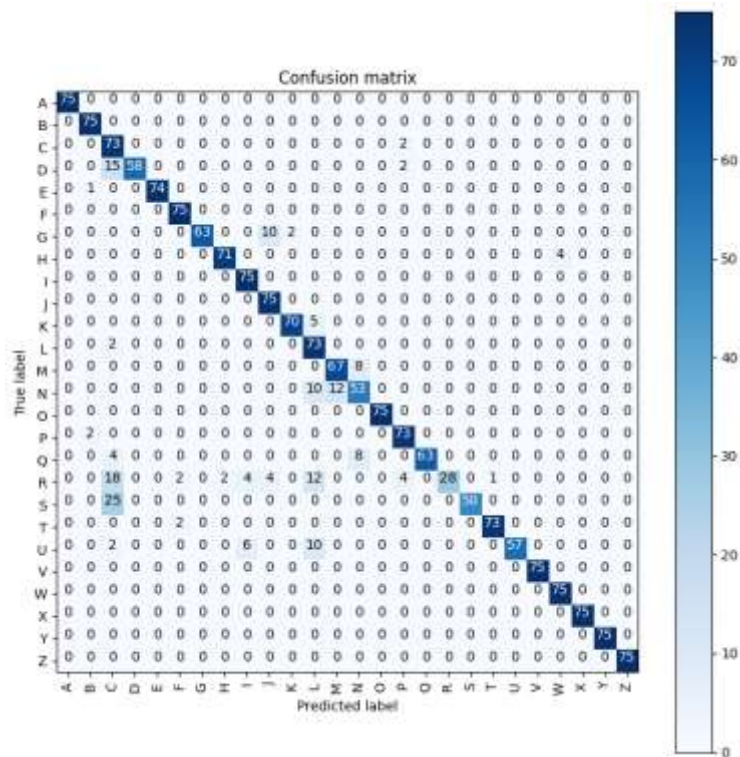
e. Evaluasi

Model yang telah dilatih dievaluasi menggunakan data validasi untuk mengukur kinerjanya. Library yang digunakan dalam evaluasi ini adalah *scikit-learn*. Dalam evaluasi model ini, tiga fungsi dari *scikit-learn* yang digunakan dalam evaluasi ini adalah *confusion_matrix*, *ConfusionMatrixDisplay*, dan *classification_report*. Proses evaluasi ini melibatkan pengukuran akurasi, presisi, recall, dan metrik lainnya untuk memastikan bahwa model bekerja dengan baik dalam mengenali huruf-huruf bahasa isyarat. *Confusion matrix* digunakan dalam mengevaluasi kinerja model klasifikasi dengan membandingkan hasil prediksi model dengan nilai sebenarnya dari data yang diuji. Untuk melihat hasil evaluasi kinerja model dapat dilihat pada Gambar 4 .3, Gambar 4 .4

```
[INFO] Testing and evaluating...
244/244 [-----] - 37s 158ms/step - loss: 0.0342 - accuracy: 0.9895
61/61 [-----] - 9s 155ms/step - loss: 0.3065 - accuracy: 0.9082
[INFO] Train accuracy: 98.95%
[INFO] Train loss: 0.034158144146283095
[INFO] Validation accuracy: 90.82%
[INFO] Validation loss: 0.3065195119380951
61/61 [-----] - 10s 155ms/step
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
precision recall f1-score support
A 1.00 1.00 1.00 75
B 0.96 1.00 0.98 75
C 0.53 0.07 0.68 75
D 1.00 0.77 0.87 75
E 1.00 0.99 0.99 75
F 0.95 1.00 0.97 75
G 1.00 0.84 0.91 75
H 0.97 0.95 0.96 75
I 0.88 1.00 0.94 75
J 0.84 1.00 0.91 75
K 0.97 0.93 0.95 75
L 0.66 0.97 0.79 75
M 0.85 0.89 0.87 75
N 0.77 0.71 0.74 75
O 1.00 1.00 1.00 75
P 0.90 0.97 0.94 75
Q 1.00 0.84 0.91 75
R 1.00 0.37 0.54 75
S 1.00 0.67 0.80 75
T 0.99 0.97 0.98 75
U 1.00 0.76 0.86 75
V 1.00 1.00 1.00 75
W 0.95 1.00 0.97 75
X 1.00 1.00 1.00 75
Y 1.00 1.00 1.00 75
Z 1.00 1.00 1.00 75

accuracy 0.93
macro avg 0.93 0.91 0.91 1950
weighted avg 0.93 0.91 0.91 1950
```

Gambar 4. 3 hasil evaluasi model



Gambar 4. 4 hasil confusion matrix.

f. Menyimpan model

Setelah melakukan semua tahapan selanjutnya model akan di simpan dalam bentuk .H5. model ini nantinya akan dijadikan API yang kan di gunakan dalam memprediksi huruf Bahasa isyarat.

3. Implementasi Sistem

Setelah melatih model dengan menggunakan CNN, tahap berikutnya adalah mendeploy model. Adapun tahapan-tahapan implementasi sistem sebagai berikut:

a. Pengumpulan Kebutuhan Sistem

1. Kebutuhan Fungsional

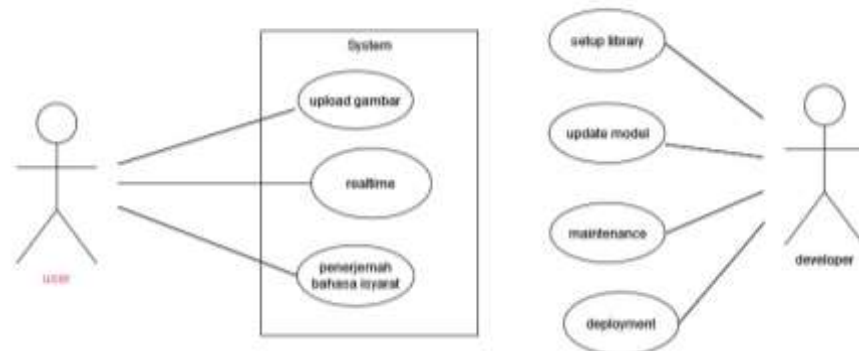
- a) Pengenalan Huruf: Aplikasi harus dapat mengenali dan memprediksi huruf-huruf bahasa isyarat yang ditampilkan melalui kamera dan upload.
- b) Interaksi Pengguna: Aplikasi harus menyediakan antarmuka yang memungkinkan pengguna untuk mengunggah gambar atau menggunakan kamera secara langsung untuk pengenalan huruf.
- c) Feedback Real-time: Aplikasi harus memberikan umpan balik secara real-time tentang hasil prediksi huruf.
- d) Penerjemah: Aplikasi dapat menerjemah dari gambar ke huruf dan juga sebaliknya

2. Desain *Unified Modelling Language* (UML)

a. Desain *Unified Modelling Language* (UML)

1) *Use Case Diagram*

Use Case Diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system.



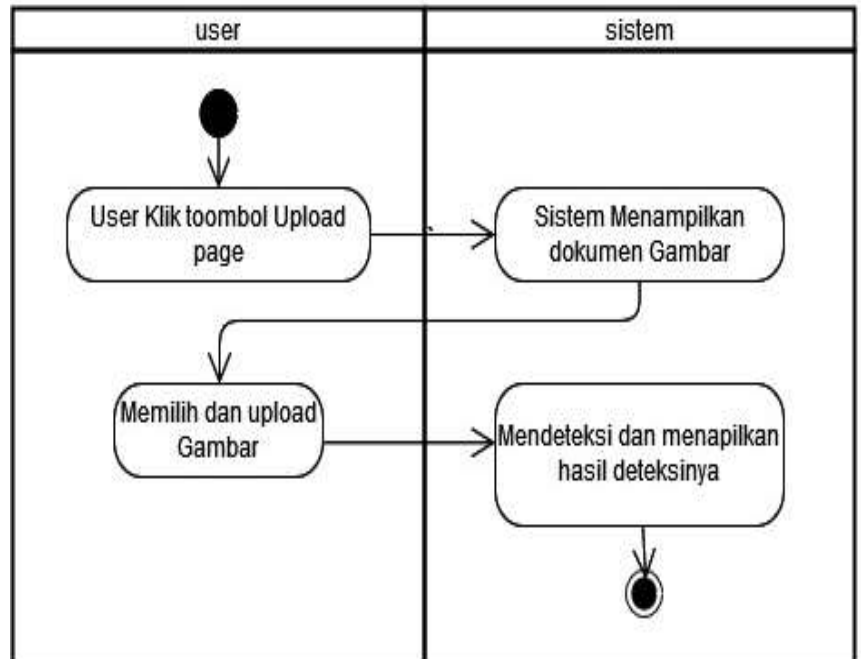
Gambar 4. 5 *use case diagram*

2) *Activity Diagram*

Activity Diagram digunakan untuk memodelkan proses bagaimana aktivitas berinteraksi satu sama lain

a. *Activity Diagram upload*

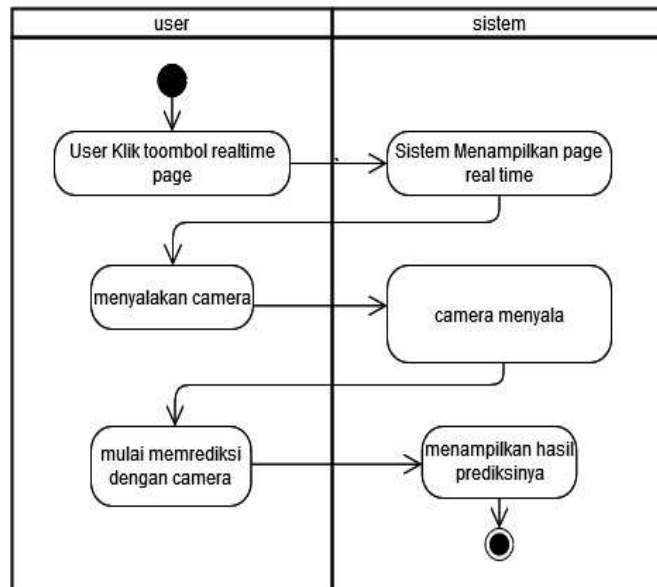
Pada *Activity Diagram upload* menjelaskan tentang memprediksi huruf abjad Bahasa isyarat dengan cara *upload*, yang merupakan salah satu fitur penting dalam *website* yang sedang dikembangkan. Pada fitur prediksi ini, pengguna diminta untuk memasukan foto dengan cara mengunggah dari penyimpanan. Jika foto telah dimasukan maka sistem akan menampilkan foto yang telah diinput selanjutnya akan diprediksi dan sistem akan menampilkan hasil prediksinya. *Activity Diagram upload* dapat dilihat pada Gambar 4. 6



Gambar 4. 6 *activity diagram upload*

b. *Activity Diagram realtime*

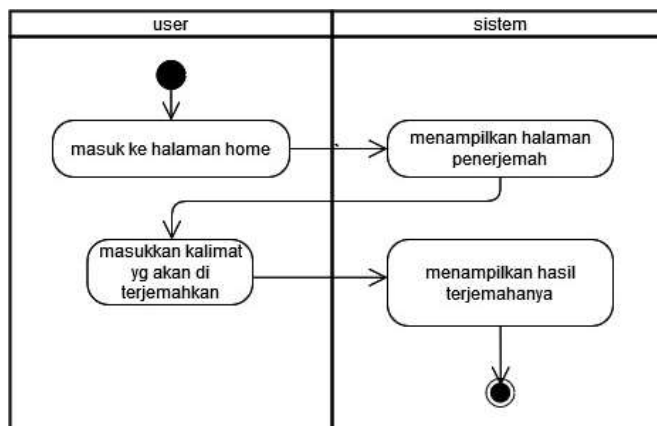
Activity Diagram realtime ini menjelaskan tentang memprediksi huruf abjad Bahasa isyarat seracar realtime atau langsung, fitur ini juga jadi salahsatu fitur penting dalam *website* yang sedang dikembangkan. Pada fitur prediksi *realtime* ini, pengguna nanti akan memeragakan Bahasa isyarat didepan kamera dan selanjutnya akan diprediksi dan sistem akan menampilkan hasil prediksinya. *Activity Diagram realtime* dapat dilihat pada Gambar 4. 7



Gambar 4. 7 activity diagram realtime

c. *Activity Diagram penerjemah*

Activity Diagram penerjemah dijelaskan merupakan fitur penerjemah di website pengenalan huruf Bahasa isyarat, pengguna dapat menerjemah kata/kalimat ke dalam gambar bahasa isyarat. Dengan cara memasukkan kalimat / kata yg ingin di terjemah maka nanti system akan menerjemah kedalam huruf bahasa isyartnya *Activity Diagram penerjemah* dapat dilihat pada Gambar 4. 8



Gambar 4. 8 Activity Diagram penerjemah

3. *Desain wireframe*

Wireframe merupakan representasi visual sederhana dari sebuah antarmuka pengguna. Adapun perancangan antarmuka yang dibuat dalam aplikasi. Berikut ini desain *wireframe* aplikasi pengenalan huruf Bahasa isyarat.

a. *Wireframe* Halaman home



Gambar 4. 9 *Wireframe* Halaman home

Pada gambar diatas merupakan halaman *home* atau halaman utama dari *website* yg akan dibangun, di gambar *wireframe* diatas terdapat beberapa kompone dari judul dan juga terdapat beberapa gambaran *wireframe* yg nantinya akan dibangun dari *wireframe* prediksi dan input teks dan juga terdapat beberapa *card view* atau kartu-kartu yang berisikan informasi tentang bahasa isyarat

b. *Wireframe* Halaman home (realtime)



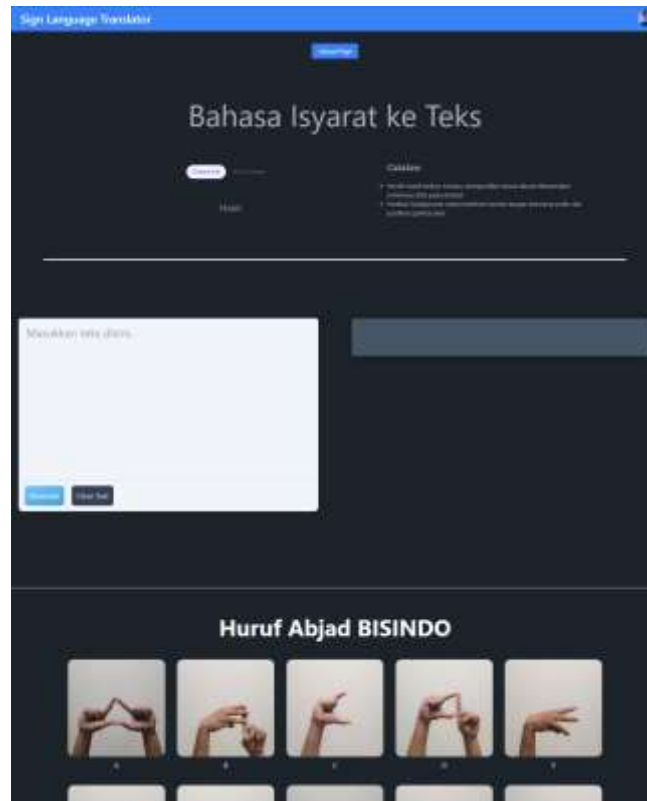
Gambar 4. 10 *Wireframe* Halaman home (realtime)

Pada gambar diatas, adalah gambaran *wireframe* dari menu home dengan fitur prediksi real time, di *wireframe* diatas terdapat gambaran fitur *real time* yang akan dibangun dari layar yg nantinya akan memunculkan camera dan nantinya juga hasil prediksinya.

b. Membangun *prototype*

Prototype dari aplikasi mulai dibangun berdasarkan kebutuhan yang telah dikumpulkan. Desain *UI/UX* dibuat untuk memastikan antarmuka yang mudah digunakan dan menarik bagi pengguna.

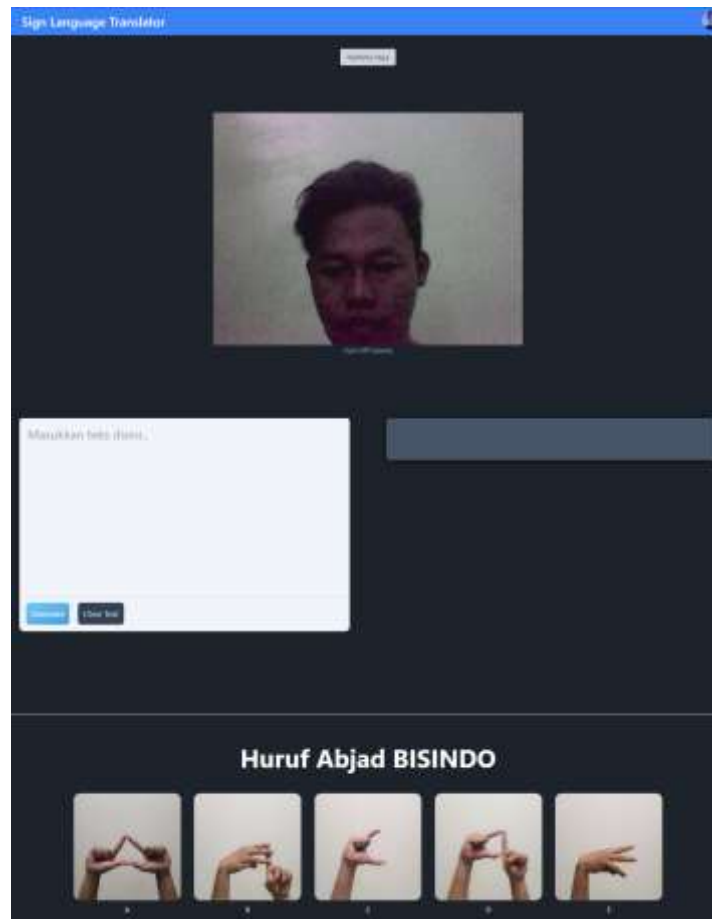
1. Halaman home



Gambar 4. 11 *prototype* halaman home

Website ini adalah alat prediksi Bahasa Isyarat Indonesia (BISINDO) ke teks. Pada halaman utamanya, pengguna disambut dengan judul "Sign Language Translator". Di bagian utama, terdapat judul "Bahasa Isyarat ke Teks" yang mengarahkan pengguna pada fitur utama aplikasi ini, yaitu memprediksi gambar bahasa isyarat. Dibawahnya juga terdapat fitur untuk menerjemah dari teks ke gambar Bahasa isyarat Di sebelah kanan, adalah nantinya hasil dari terjemahannya. Kolom teks di sebelah kiri adalah form untuk memasukkan kata yang ingin di terjemah, seperti contoh kata "hallo". Di bagian bawah, terdapat tampilan huruf abjad BISINDO, yang menunjukkan setiap huruf dalam alfabet BISINDO dengan ilustrasi gerakan tangan yang sesuai.

2. Halaman home (realtime)



Gambar 4. 12 *prototype halaman home realtime*

Halaman ini merupakan bagian dari website prediksi Bahasa Isyarat Indonesia (BISINDO) ke teks. Pada bagian atas halaman, terdapat judul "*Sign Language Translator*". Di tengah halaman, terdapat area video yang menampilkan gambar pengguna saat ini, lengkap dengan tombol "*Turn Off Camera*" di bawahnya untuk mematikan kamera jika diperlukan. Area video ini digunakan untuk menangkap gerakan bahasa isyarat yang akan diterjemahkan menjadi teks.

c. *Evaluasi prototype*

Evaluasi prototype adalah tahap penting dalam pengembangan sistem untuk memastikan hasil akhir sesuai dengan harapan dan kebutuhan klien. Melalui evaluasi yang teliti dan umpan balik yang konstruktif, prototype dapat disempurnakan sebelum memasuki tahap

berikutnya. Dari evaluasi yang telah dilakukan, terdapat beberapa perbaikan yang diusulkan. Pertama, judul *header* perlu menggunakan font yang lebih tegas dan ukuran yang lebih besar untuk meningkatkan keterbacaan dan visibilitas dan juga judul di ubah ke Bahasa Indonesia agar mudah di pahami. Kedua, tombol on/off kamera perlu didesain ulang agar terlihat lebih jelas dan mudah dikenal. Selain itu, pemilihan warna dan penempatan tombol harus strategis agar mudah ditemukan dan diakses oleh pengguna. Dengan memperbaiki aspek-aspek tersebut, diharapkan prototype aplikasi menjadi lebih ramah pengguna dan efektif dalam memenuhi kebutuhan pengguna.

d. Mengkodekan Sistem

Selanjutnya pengkodean sistem secara penuh menggunakan *React.js* untuk *front-end* dan *Flask* untuk *back-end*. Integrasi dengan model pengenalan huruf bahasa isyarat juga dilakukan pada tahap ini.

1. Pembuatan *Flask*

Pada pengembangan web ini menggunakan *Flask* sebagai *framework* untuk *back-end*. *Flask* memungkinkan kita untuk membuat aplikasi *web* dengan cepat dan mudah. Dengan menggunakan *Flask*, kita dapat mengatur *rute (route)* untuk menangani permintaan *HTTP*, seperti *GET* dan *POST*, serta mengelola berbagai respons yang diperlukan oleh aplikasi. Disini *flask* hanya digunakan untuk pembuatan *API* dengan metode *POST* berikut adalah tahapannya:

a. *Requirement*

Proses pembuatan *API* ini menggunakan berbagai modul tersebut untuk membuat aplikasi web dengan Flask, memuat model neural network, memproses gambar, dan menghasilkan respons *JSON* berdasarkan prediksi model berikut *requirement* yang dipakai untuk pembuatan flask dapat dilihat Tabel 4.1

Tabel 4. 3 *Requirement*

No	Nama	Penjelasan
1	<i>Flask</i>	Framework web untuk membuat API
2	<i>Request</i>	Modul untuk menangani permintaan HTTP.
3	<i>Jsonify</i>	Fungsi untuk mengubah objek Python menjadi respon JSON.
4	<i>Tensorflow.keras.models</i>	Sub-modul dari TensorFlow untuk bekerja dengan model neural network.
5	<i>load_model</i>	Fungsi untuk memuat model Keras yang telah dilatih.
6	<i>flask_cors</i>	Modul untuk mengatasi masalah Cross-Origin Resource Sharing (CORS) dalam aplikasi Flask.
7	<i>numpy</i>	Library untuk melakukan operasi matematika dengan array dan matriks.
8	<i>PIL</i>	Library untuk melakukan operasi matematika dengan array dan matriks.
9	<i>io</i>	Modul input-output standar Python, digunakan dalam hal ini untuk membaca gambar yang diterima dalam format byte.

b. Inisialisasi Aplikasi *Flask* dan Pemodelan.

Pada bagian ini, aplikasi *Flask* diinisialisasi dan model TensorFlow yang telah dilatih dimuat dari file `model_new4.h5`. *CORS (app)* digunakan untuk mengizinkan permintaan dari domain lain, yang penting untuk mengatasi *masalah Cross-Origin Resource Sharing*.



```

1 from flask import Flask, request, jsonify
2 from tensorflow.keras.models import load_model
3 from flask_cors import CORS
4 import numpy as np
5 from PIL import Image
6 import io
7
8 app = Flask(__name__)
9 model = load_model('model_new4.h5')
10 CORS(app)

```

Gambar 4. 13 Inisialisasi Aplikasi *Flas*

c. Definiskan Label Kelas

Class_labels adalah daftar label yang sesuai dengan output dari model. Setiap indeks dalam prediksi model akan berkorespondensi dengan label ini.



```

1 # Definiskan Label kelas Anda
2 class_labels = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y"]
3

```

Gambar 4. 14 Label Kelas

d. Rute untuk Prediksi Gambar

Rute untuk prediksi gambar adalah bagian penting dari aplikasi web ini. Pada bagian ini mengirimkan gambar melalui permintaan *POST* ke *endpoint /predict*. Langkah-langkahnya meliputi mengubah ukuran gambar, mengonversi gambar menjadi format yang dapat diterima oleh model, dan melakukan

normalisasi. Setelah itu, gambar yang telah diproses dimasukkan ke dalam model untuk melakukan prediksi. Hasil prediksi, berupa label kelas yang paling mungkin, dikembalikan sebagai respons *JSON*. Dengan begitu, rute ini memungkinkan Anda untuk mengunggah gambar dan menerima prediksi kelas dari model yang telah dilatih dengan *Flask*.



```

1  @app.route('/predict', methods=['POST'])
2  def predict_image():
3      file = request.files['image']
4      img = Image.open(io.BytesIO(file.read()))
5      img = img.resize((224, 224))
6      img_array = np.asarray(img)
7      img_array = img_array / 255.0
8      img_array = np.expand_dims(img_array, axis=0)
9      prediction = model.predict(img_array)
10     predicted_index = np.argmax(prediction[0])
11     predicted_class = class_labels[predicted_index]
12
13     return jsonify({'prediction': predicted_class})
14

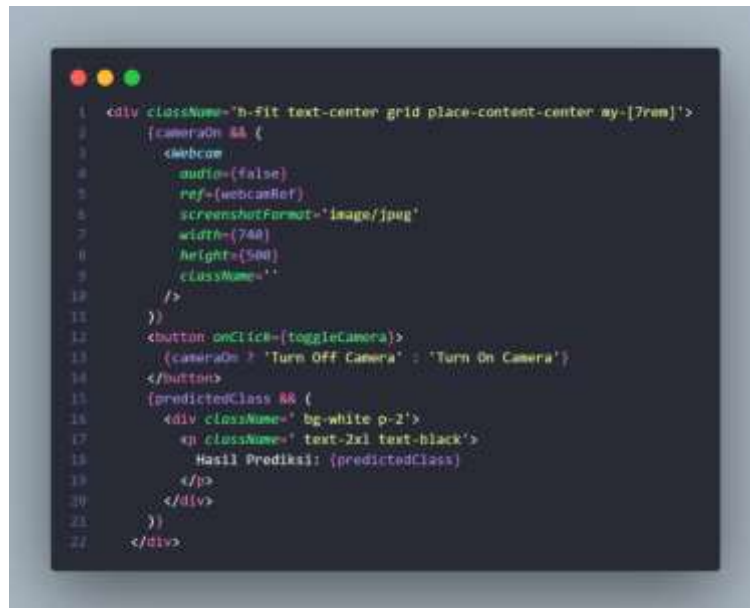
```

Gambar 4. 15 rute prediksi

2. *React.js*

Pada pengembangan web ini menggunakan *React.js* sebagai *framework* untuk *front-end*. *React.js* adalah pustaka *JavaScript* yang populer untuk membangun antarmuka pengguna yang dinamis. *React* memungkinkan pengembang untuk membangun aplikasi web yang kompleks dengan lebih mudah dan efisien. Berikut adalah potongan code pembuatan *front-end* menggunakan *React.js*.

b. Deteksi *realtime*



```

1 <div className="h-fit text-center grid place-content-center my-[7rem]">
2   {cameraOn && (
3     <webcam
4       audio={false}
5       ref={webcamRef}
6       screenshotFormat="image/jpeg"
7       width={740}
8       height={500}
9       className=""
10      />
11    )}
12   <button onClick={toggleCamera}>
13     {cameraOn ? "Turn Off Camera" : "Turn On Camera"}
14   </button>
15   {predictedClass && (
16     <div className="bg-white p-2">
17       <p className="text-2xl text-black">
18         Hasil Prediksi: {predictedClass}
19       </p>
20     </div>
21   )}
22 </div>

```

Gambar 4. 17 deteksi realtime

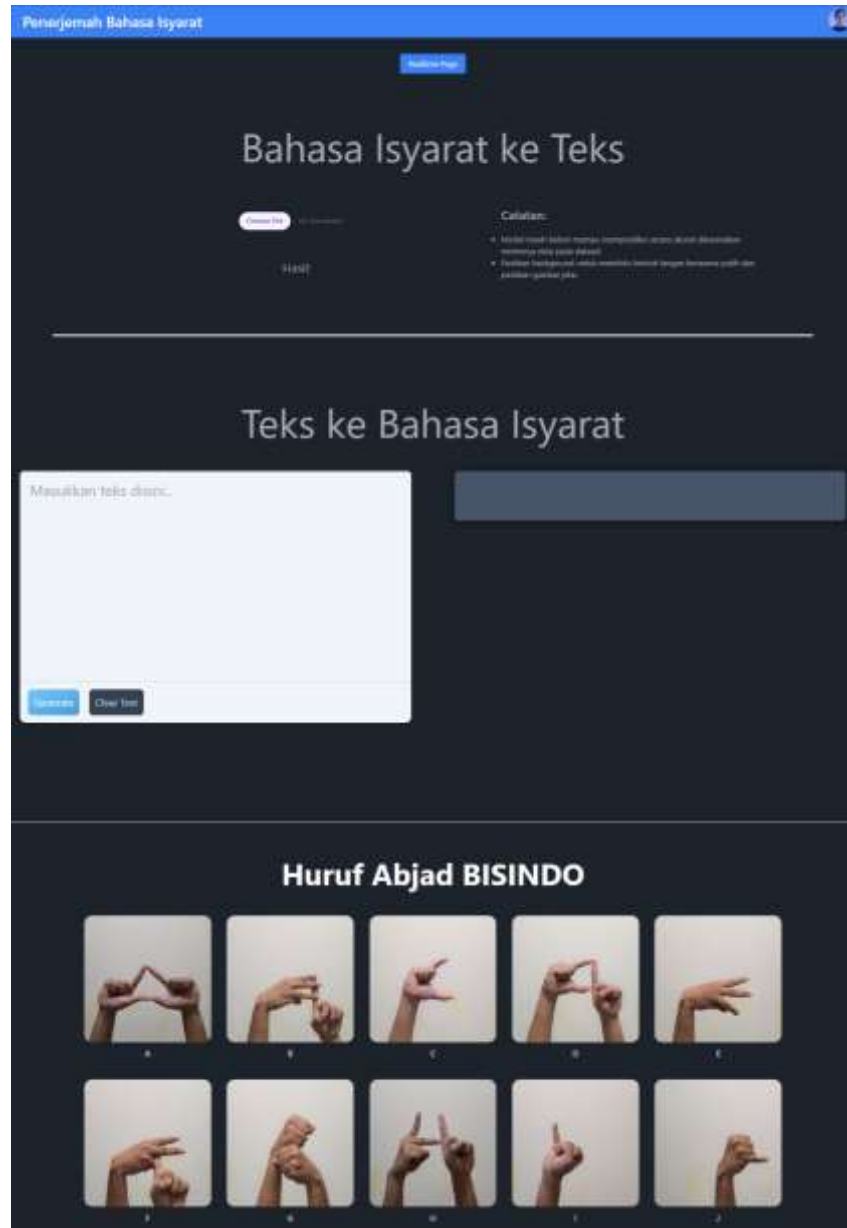
Komponen *Realtime* di *React* memungkinkan prediksi *real-time* dari webcam menggunakan model pembelajaran mesin. *State predictedClass* menyimpan hasil prediksi dan *cameraOn* mengatur status kamera. mengirimnya ke *server*, dan menerima prediksi, diulang menggunakan *requestAnimationFrame*. *useEffect* memastikan prediksi berjalan selama kamera menyala.

e. Hasil *website*

1. Halaman Home

Beranda adalah halaman pertama yang muncul saat membuka website. Pada halaman utamanya terdapat bagian yang mengarahkan pengguna pada salah satu fitur utama aplikasi ini, yaitu menerjemahkan / deteksi bahasa isyarat menjadi teks. Pertama terdapat salah satu fitur untuk memprediksi Bahasa isyarat yaitu berupa upload gambar, gambar yg di upload nantinya akan diprediksi dan akan memunculkan hasil prediksinya ,dan juga dihalam pertama ini juga terdapat fitur dengan menerjemah dari tulisan/teks ke

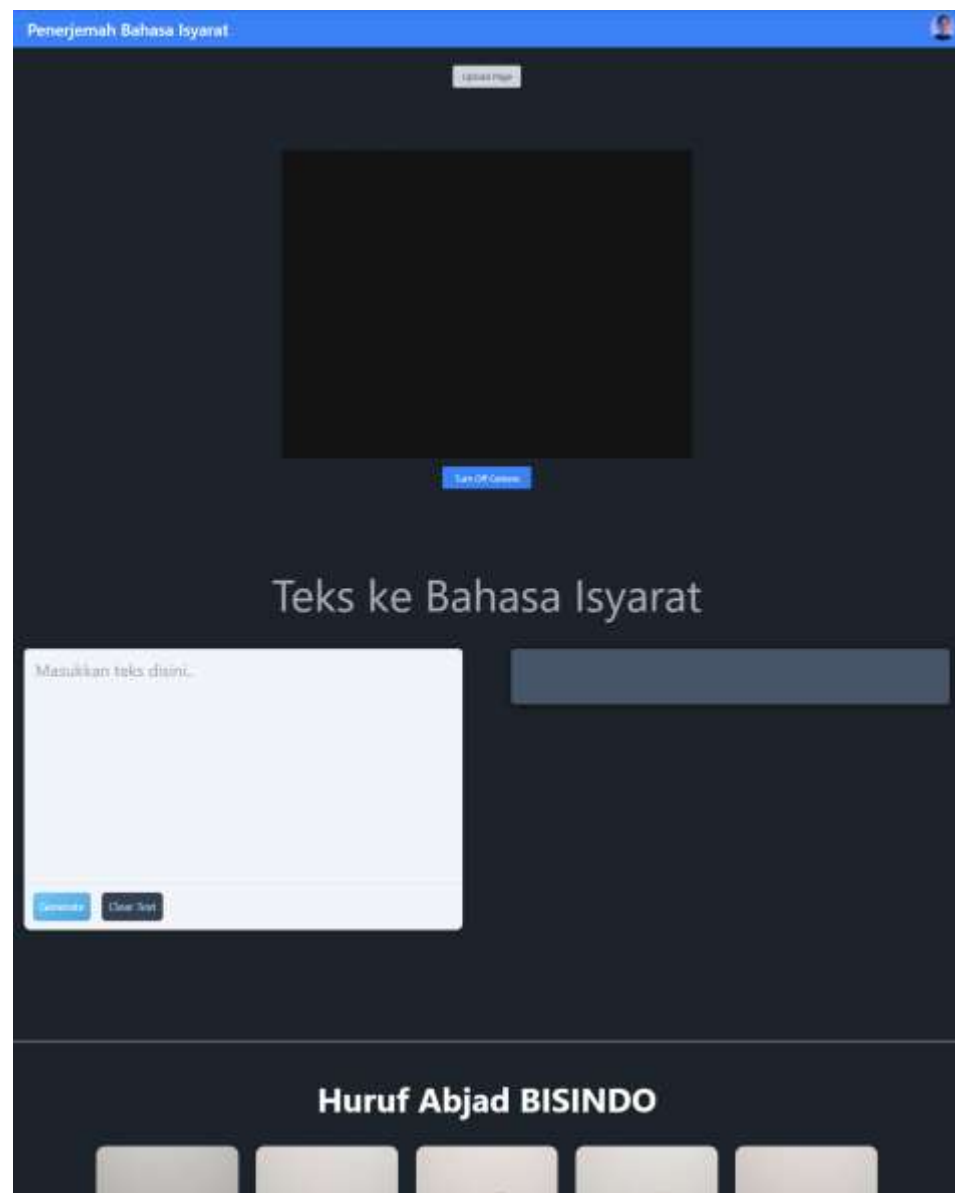
gambar huruf Bahasa isyarat. Di bagian bawah, terdapat tampilan huruf abjad BISINDO, yang menunjukkan setiap huruf dalam alfabet BISINDO dengan ilustrasi gambar tangan yang sesuai.



Gambar 4. 18 hasil website halaman home

2. Halaman home (realtime)

Di halaman ini adalah menampilkan salah satu fitur dari website yang sudah di bangun yaitu memprediksi melalui camera atau memprediksi secara realtime. Di tengah halaman, terdapat area camera yang menampilkan gambar pengguna saat ini, Area camera digunakan untuk menangkap gerakan bahasa isyarat yang akan diterjemahkan menjadi teks.



Gambar 4. 19 hasil website halaman home realtime

f. Menguji system (*white box dan black box*)

Pengujian dilakukan untuk memastikan bahwa aplikasi yang sudah dibuat berfungsi dengan baik. Dalam penelitian ini, menggunakan 2 jenis pengujian yaitu *white box testing*, dan *black box testing*. Rincian pengujian yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. *White Box Testing*

White Box Testing adalah metode pengujian perangkat lunak di mana penguji meneliti apakah kode yang dibuat terdapat kesalahan atau tidak dan struktur internal dari aplikasi yang diuji. Berikut hasil pengujian white box dapat dilihat pada Tabel 4.4

Tabel 4. 4 *White Box Testing*

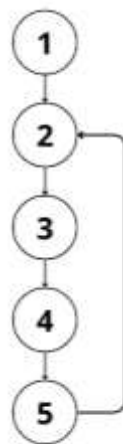
Node	Code	Penegrtian
1.	<pre>ReactDOM.createRoot (document.getElementById('root')).render(<React.StrictMode> <App /> </React.StrictMode>,)</pre>	Menampilkan halaman untuk prediksi
2.	<pre><form className='space-x-6'> <div className='shrink-0'></div> <label className='block'> <input type='file' className='block w-full text-sm text-slate-500 file:mr-4 file:py-2 file:px-4 file:rounded-full file:border-0 file:text-sm file:font-semibold file:bg-violet-50 file:text-violet-700 hover:file:bg-violet-100 hover:cursor- pointer py-10'</pre>	<i>Form input gambar</i>

	<pre> name='image' onChange={handleImageAndPredict} /> </label> </form> </pre>	
3.	<pre> useEffect(() => { if (!selectedFile) { setPreview(undefined) return } const objectUrl = URL.createObjectURL(selectedFile) setPreview(objectUrl) return () => URL.revokeObjectURL(objectUrl) }, [selectedFile]) </pre>	<p><i>Image preview gambar yang akan di prediksi</i></p>
4.	<pre> const handleImageAndPredict = async e => { if (e.target.files.length === 0) { setPredictedClass(null) } if (e.target.files.length === 1) { setLoading(true) const file = e.target.files[0] setSelectedFile(file) const formData = new FormData() formData.append('image', file) try { const response = await fetch('http://127.0.0.1:5000/predict', { </pre>	<p>Proses prediksi gambar</p>

	<pre> method: 'POST', body: formData, }) const result = await response.json() setPredictedClass(result.prediction) } catch (error) { console.error('Error making prediction:', error) } setLoading(false) } } </pre>	
5.	<p><p className='font-normal text-2xl p-5'>Hasil: {predictedClass}</p></p>	Menampilkan hasil prediksi

a) Membuat *Flowgraph* dari Program.

berikut adalah gambaran *Flowgraph* dari script penilaian diatas dapat dilihat pada Gambar 4.20



Gambar 4. 20 *Flowgraph*

b) Menghitung Kompleksitas Siklomatik.

Kompleksitas siklomatis adalah metrik yang digunakan untuk mengukur kompleksitas struktural dari sebuah program.

Rumus untuk menghitung kompleksitas siklomatik:

$$V(G) = E - N + 2$$

Dimana:

- E adalah jumlah edge (garis alur dalam flow graph)
- N adalah jumlah node (titik keputusan dalam flow graph)

Berikut adalah perhitungan Kompleksitas Siklomatik diperoleh dengan Perhitungan berikut:

Nodes: 5

Edges: 5

Menggunakan rumus kompleksitas siklomatik:

$$V(G) = E - N + 2 = 5 - 5 + 2 = 2$$

Dari hasil perhitungan diatas dapat ditentukan independent path sebagai berikut Jalur Independen. Terdapat 2 independent path yang diperoleh. Berikut dibawah ini :

Path 1 : 1, 2, 3, 4, 5,

Path 2 : 1, 2, 3, 4, 5, 2

2. *Black Box Testing*

Black box testing digunakan untuk mengevaluasi kinerja aplikasi tanpa memeriksa struktur internalnya, dengan tujuan memastikan bahwa aplikasi berfungsi seperti yang diharapkan oleh pengguna. Proses pengujian ini terstruktur dalam tiga tahap kunci: perencanaan, pelaksanaan, dan evaluasi. Pengujian ini dilakukan oleh 3 dosen informatika, fokus utama dari pengujian ini adalah untuk menilai sejauh mana aplikasi memenuhi fungsionalitas yang diinginkan oleh pengguna akhir

a) Hasil Perhitungan *Black Box Testing*

Hasil Perhitungan *Black Box Testing* dapat dilihat pada Tabel 4.5

Tabel 4. 5 hasil *Black Box Testing*

Nama pengujian	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	keterangan					
				diterima			ditolak		
				1	2	3	1	2	3
Halaman home	User dapat melakukan gulir atau scroll kebawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi (upload), form input teks ke bahasa isyarat, dan daftar huruf Bahasa isyarat bisindo)	Website akan menampilkan semua fitur yang ada pada halaman home	√	√	√			
	User dapat menekan tombol realtime page untuk berganti ke page realtime	User dapat melihat bagian realtime page untuk prediksi secara realtime	Website menampilkan page bagian realtime untuk prediksi	√	√	√			

Halaman home (prediksi upload)	User dapat menekan tombol choose file	User dapat memilih gambar yang akan di upload untuk di prediksi dari file manager	Website akan menampilkan file manager, menampilkan preview dari gambar yang akan di prediksi, dan akan menampilkan hasil prediksi	√	√	√			
Halaman home (prediksi realtime)	User dapat mengarahkan objek ke kamera untuk di prediksi	User dapat melihat hasil prediksi realtime	Website menampilkan hasil prediksi realtime	√	√	√			
	User dapat menekan tombol on/of camera	User dapat mengaktifkan dan menonaktifkan kamera	Website dapat menampilkan hasil ketika kamera aktif dan non aktif	√	√	√			

Halaman home (teks ke Bahasa isyarat)	User dapat melakukan terjemah dari teks ke Bahasa isyarat	User dapat memasukan keyword teks yang akan di terjemah ke bahasa isyarat	Website dapat memunculkan terjemahan hasil	√	√	√			
---------------------------------------	---	---	--	---	---	---	--	--	--

b) Kesimpulan Hasil *Black Box Testing*

Berdasarkan hasil pengujian *black box* yang sudah dilakukan dari 6 *Test Case website* yang sudah dibuat di ujikan kepada 3 dosen informatika berikut adalah hasilnya:

1) Pengujian pertama

$$\text{Tercapai} : \frac{6}{6} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{6} \times 100 \% = 0 \%$$

2) Pengujian kedua

$$\text{Tercapai} : \frac{6}{6} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{6} \times 100 \% = 0 \%$$

3) Pengujian ketiga

$$\text{Tercapai} : \frac{6}{6} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{6} \times 100 \% = 0 \%$$

$$\text{Jumlah presentase rata-rata tercapai} = \frac{300\%}{3} = 100\%$$

$$\text{Jumlah presentase rata-rata gagal} = \frac{0\%}{3} = 0\%$$

Berdasarkan penghitungan di atas, bahwa dari 6 pengujian

oleh 3 responden menunjukkan tingkat keberhasilan pengujian black box memiliki presentase 100%, sedangkan yang gagal memiliki presentase 0%. Disini dapat disimpulkan bahwa website yang sudah dibuat berjalan sesuai fungsionalitasnya.

g. Evaluasi *User Acceptance Testing* (UAT)

Setelah pengujian teknis, aplikasi diuji oleh pengguna akhir untuk memastikan bahwa semua kebutuhan telah terpenuhi dan aplikasi siap untuk digunakan secara luas. Pengujian *User acceptance testing* dilakukan untuk memastikan bahwa aplikasi klasifikasi sampah organik dan anorganik memenuhi kebutuhan dan harapan pengguna. *User acceptance testing* diujikan pada 5 responden.

Berikut ini merupakan hasil kuesioner pengujian user acceptance testing yang telah disebarakan kepada 5 responden. Hasil pengujian user acceptance testing dapat dilihat pada Tabel 4.6

Tabel 4. 6 hasil UAT

pertanyaan	Hasil Pengujian				
	Responden 1	Responden 2	Responden 3	Responden 4	Responden 5
1	5	5	5	5	5
2	5	5	5	5	5
3	5	5	5	5	5
4	5	4	5	5	4
5	5	5	5	5	4
6	3	4	5	5	5
7	5	5	5	5	5
8	5	5	5	5	4
9	4	3	5	5	4
10	4	5	5	5	5

Jumlah skor	46	46	50	50	46
Presentase	92%	92%	100%	100%	92%
Total	476%				

Dari hasil presentase setiap pertanyaan kemudian dicari nilai rata-ratanya untuk mencapai tingkat penerimaan terhadap aplikasi yang dibuat oleh responden. Nilai rata-ratanya dapat dihitung dengan menggunakan persamaan sebagai berikut :

$$\text{Presentase rata-rata} : \frac{\text{jumlah total presentasi}}{\text{jumlah responden}}$$

$$\text{Presentase rata-rata} : \frac{476\%}{5} = 95.2\%$$

Dengan daftar kategori presentase sebagai berikut:

0% - 20% = Sangat Kurang

21% - 40% = Kurang

41% - 60% = Cukup Baik

61% - 80% = Baik

81% - 100% = Sangat Baik

Dari perhitungan tersebut, diperoleh presentase rata-rata sebesar 95.2%. Dengan demikian, dapat disimpulkan bahwa pengujian UAT pada aplikasi ini mendapat kategori "Sangat Baik".

B. Pembahasan

1. Analisis Kebutuhan Sistem

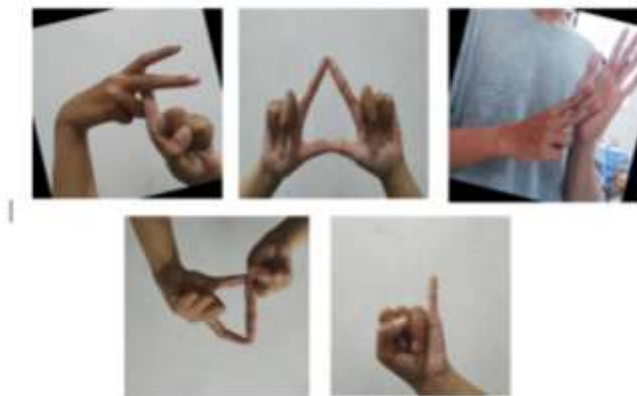
Analisis kebutuhan sistem adalah langkah pertama dalam pengembangan aplikasi pengenalan huruf bahasa isyarat. Dalam pengembangan aplikasi pengenalan huruf bahasa isyarat. Aplikasi ini membutuhkan perangkat keras dengan spesifikasi minimum seperti

processor AMD E2 7th dan RAM 4GB untuk backend, serta Windows 64-bit, RAM 4GB, dan resolusi layar 1280 x 800 untuk frontend.

Perangkat lunak yang dibutuhkan termasuk *Google Colaboratory* untuk pengembangan machine learning, *Visual Studio Code* untuk pengkodean, *Github* untuk manajemen kode, *Postman* untuk pengujian API, *Figma* untuk desain UI/UX, *Kaggle* untuk dataset dan eksperimen ML, serta *Google Drive* untuk penyimpanan file proyek. Dengan memenuhi kebutuhan perangkat keras dan lunak ini, pengembangan aplikasi dapat dilakukan dengan lancar dan efisien, memastikan setiap aspek sistem berfungsi optimal dalam mendukung aplikasi pengenalan huruf bahasa isyarat.

2. Implementasi Model Pengenalan Huruf Bahasa Isyarat

Proses pengembangan model untuk pengenalan huruf bahasa isyarat dimulai dengan pengumpulan dataset dari Kaggle, yang terdiri dari gambar huruf A-Z dibagi menjadi *training data* (80%) dan *validation data* (20%). Selanjutnya dataset yang sudah diunduh dilakukan *handel imbalance dataset* agar model yang dihasilkan tidak menjadi model yang bias. Dataset kemudian diolah melalui *augmentasi* gambar untuk meningkatkan variasi dan kuantitas data pelatihan dengan teknik seperti *Rotation Range*, *Width Shift Range*, *Height Shift Range*, *Horizontal Flip*. Gambar hasil augmentasi bis dilihat pada gambar 4. 21



Gambar 4. 21 hasil augmentasi

Pada tahapan pembuatan model ini pada awalnya menggunakan CNN murni dikarenakan pada hasil model yang dihasilkan kurang bagus dan mengalami *overfitting* maka proses pembuatan modelnya diganti menggunakan *Transfer Learning* dengan arsitektur *MobileNetV2*. Model yang dibangun menggunakan *Transfer Learning* dengan arsitektur *MobileNetV2* ini, dimana lapisan pra-pelatihan disesuaikan dengan menambahkan lapisan *Global Average Pooling*, lapisan *dense* (1024, 512, 256 *neuron*) dengan *dropout* 0.2, dan lapisan *output* dengan aktivasi *Softmax* untuk mengklasifikasikan 26 kelas huruf isyarat. Proses pelatihan model dilakukan dengan kompilasi menggunakan *optimizer Adam*, fungsi *loss categorical_crossentropy*, dan juga pada saat pelatihan ini menggunakan *epoch* sebanyak 50 dan juga menggunakan *callback* jenis *early stop*, *callback* ini digunakan digunakan untuk menghentikan proses pelatihan jika akurasi model mencapai lebih dari 96%, hasil training dapat di lihat dalam gambar 4. 22

```

Epoch 1/50
244/244 [=====] - 85s 229ms/step - loss: 3.1119 - accuracy: 0.1173 - val_loss: 2.6737 - val_accuracy: 0.3398
Epoch 2/50
244/244 [=====] - 54s 221ms/step - loss: 1.4886 - accuracy: 0.5622 - val_loss: 0.9377 - val_accuracy: 0.7241
Epoch 3/50
244/244 [=====] - 55s 223ms/step - loss: 0.3584 - accuracy: 0.8986 - val_loss: 0.6690 - val_accuracy: 0.8388
Epoch 4/50
244/244 [=====] - ETA: 0s - loss: 0.1171 - accuracy: 0.9662Accuracy above 97% -- Stop Training
244/244 [=====] - 53s 215ms/step - loss: 0.1171 - accuracy: 0.9662 - val_loss: 0.3965 - val_accuracy: 0.9082

```

Gambar 4. 22 hasil trainig

tujuan dari penggunaan *callback* ini adalah agar model tidak *overfitting*, selanjutnya adalah evaluasi menggunakan data validasi untuk mengukur kinerja model. Evaluasi dilakukan dengan memonitor akurasi dan metrik lainnya seperti *presisi* dan *recall*, sambil menggunakan *confusion matrix* untuk mengevaluasi hasil prediksi model terhadap data validasi dan yang terakhir adalah menyimpan model dalam bentuk *.H5* yang nantinya akan digunakan dalam pembuatan API dalam implementasi sistem.

3. Implementasi Sistem

Setelah melatih model menggunakan CNN, tahap selanjutnya adalah mendeploy model tersebut. Impelemntasi system ini menggunakan metode

prototyping, Pada proses metode *prototyping* ini untuk pengembangan website, dimulai dengan pengumpulan kebutuhan sistem berupa perangkat lunak dan keras. Selanjutnya, prototipe awal berupa desain *UI/UX* dibuat dan diuji oleh sejumlah kecil pengguna untuk mendapatkan umpan balik. Setelah itu, sistem dikodekan menggunakan *React JS* dan *Flask*. Proses pengujian dilakukan dengan metode *Black Box*, *White Box*, pengujian *arsitektur*, dan *Basis Path* untuk memastikan fungsi website berjalan baik. Tahap akhir adalah evaluasi melalui UAT oleh klien untuk memastikan kesesuaian *website* dengan kebutuhan mereka. Implementasi sistem dimulai dengan pengumpulan kebutuhan fungsional, termasuk kemampuan aplikasi untuk mengenali huruf bahasa isyarat, menyediakan antarmuka pengguna untuk mengunggah gambar atau menggunakan kamera, memberikan *feedback real-time*, dan menerjemahkan gambar ke huruf atau sebaliknya. Desain sistem menggunakan *Unified Modelling Language (UML)*, termasuk *Use Case Diagram*, *Activity Diagram* untuk berbagai aktivitas, dan *wireframe* untuk tampilan antarmuka pengguna. Prototype aplikasi dikembangkan dengan desain *UI/UX* yang menarik dan mudah digunakan. Dari pembuatan prototipe ini dilakukannya evaluasi diuji oleh sejumlah kecil pengguna, beberapa perbaikan diusulkan. Pertama, judul header perlu menggunakan font yang lebih tegas dan ukuran yang lebih besar untuk meningkatkan keterbacaan, serta mengubah judul ke Bahasa Indonesia agar lebih mudah dipahami. Kedua, tombol on/off kamera perlu didesain ulang agar terlihat lebih jelas dan mudah dikenali, dengan pemilihan warna dan penempatan tombol yang strategis agar mudah ditemukan dan diakses oleh pengguna. Dengan perbaikan ini, diharapkan prototype menjadi lebih ramah pengguna dan efektif.

Pengkodean sistem dilakukan dengan *React.js* untuk *front-end* dan *Flask* untuk *back-end*, di mana *Flask* mengatur *API* dengan berbagai modul seperti *Flask*, *Request*, *Jsonify*, *TensorFlow*, *numpy*, *PIL*, dan *io*. Aplikasi *Flask* diinisialisasi dan model *TensorFlow* dimuat, dengan definisi label kelas dan rute untuk prediksi gambar melalui endpoint */predict*. *Front-end* menggunakan *React js*, *React.js* sendiri adalah salah satu *framework* dari

javascript penulis menggunakan *react js* sendiri dikarenakan pada awalnya model yang sudah dibuat akan di *convert* menjadi *tfjs* agar bisa digunakan dalam *react js*, dikarenakan hasil dari *convert* ke *tfjs* mengalami penurunan akurasi maka penggunaan dari *tfjs* ini diganti dengan menggunakan API yang dibuat menggunakan *Flask*. Fitur yang dihasilkan dari *react js* ini adalah fitur untuk upload gambar dan deteksi *real-time* dari *webcam*, agar memudahkan interaksi pengguna dan pengiriman data ke server. Hasil *website* mencakup halaman *home* dengan fitur upload gambar dan penerjemahan teks, serta halaman *real-time* yang menampilkan area kamera untuk menangkap gerakan bahasa isyarat.

Tahap terakhir adalah pengujian untuk memastikan kualitas aplikasi yang berfungsi dengan baik. Pengujian ini dilakukan dengan 3 pengujian yaitu pengujian *Black Box*, pengujian *White Box*, dan pengujian *User Acceptance Testing (UAT)*. Pengujian *White Box* memberikan hasil yang memuaskan dengan presentase 100% dengan 2 path keberhasilan. Dalam pengujian *Black Box* mendapatkan hasil presentase sebanyak 100% pengujian berhasil, sedangkan dengan pengujian yang gagal mendapatkan presentase 0% dari 3 responden yaitu dosen informatika dengan 6 pengujian. Pengujian *user acceptance test (UAT)* juga berhasil dengan presentase 95.2% dari 5 responden dengan 10 pertanyaan pada pengujian UAT ini diujikan langsung pada orang yang tuna rungu dan juga guru slb. Untuk pengujian yang pertama UAT ini diujikan kepada salah satu mahasiswa upgris penyandang tuna rungu hasil pengujiannya mendapatkan presentase 92%, yang kedua yaitu juga penyandang tuna rungu dengan hasil presentase 92%, yang ketiga pengujian ini diujikan kepada salah satu guru SLB YPAC semarang dengan hasil presentasi pengujian sebesar 92%, dan yang terakhir 2 responden diujikan ke masyarakat umum dnegan hasil presentase pengujian 100%.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan penulis, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dari hasil proses pelatihan model untuk pengenalan huruf abjad Bahasa isyarat mengunkana metode CNN menghasilkan hasil accuracy sebesar 0.96%, validation accuracy sebesar 0.90%. pelatihan model ini menggunakan teknik *tranfer learning MobileNetV2* dengan dataset *training* sebanyak 7.800 (80%) dan *validasi* sebnayak 1950 (20%).
2. Penelitian ini menggunakan 3 tahapan pengujian pada pengembangan aplikasi, yaitu pengujian *black box*, pengujian *white box*, dan pengujian *user acceptance test (UAT)* dan menghasilkan hasil yang bagus. Pada pengujian *black box* mendapatkan hasil presentase sebanyak 100% pengujian berhasil, sedangkan dengan pengujian yang gagal mendapatkan presentase 0% dari 3 responden dalam 6 pengujian. Pengujian *white box* juga mencapai keberhasilan 100%, dan pengujian yang terakhir pengujian *User Acceptance Test (UAT)* mendapatkan hasil baik dengan presentase sebanyak 95.2% dari 5 responden dengan 10 pertanyaan.
3. Penerapan metode *Convolutional Neural Network* terbukti berhasil dalam mengenali huruf-huruf bahasa isyarat. Uji coba menunjukkan bahwa model CNN memiliki tingkat akurasi yang baik dalam mengidentifikasi gambar isyarat huruf.

B. SARAN

1. Pertimbangkan untuk memperluas dataset dengan menambahkan lebih banyak gambar yang beragam. Gunakan variasi pencahayaan, latar belakang, dan posisi tangan untuk membantu model belajar dengan lebih baik.
2. gunakan arsitektur CNN yang lebih canggih atau manfaatkan teknik *transfer learning* yang lain selain *MobileNetV2* dari model yang sudah dilatih sebelumnya untuk meningkatkan kinerja model.
3. Dalam pengembangan sistem selanjutnya diharapkan bisa berbasis mobile android agar dapat mempermudah dalam pengoperasiannya dapat diakses dimanapun dan kapanpun.

DAFTAR PUSTAKA

- [1] A. Nugroho, "DETEKSI BAHASA ISYARAT BISINDO MENGGUNAKAN METODE MACHINE LEARNING" *Processor: Jurnal Ilmiah Sistem Informasi, Teknologi Informasi*, vol. 18, no. 2, pp. 152-158, 2023.
- [2] Z. N. A. A. Ichwanul Muslim Pramono, "MODEL PENERJEMAH BAHASA ISYARAT INDONESIA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)," *Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK)*, pp. 1-5, 2024.
- [3] J. T. Sihotang, "SETARA (APLIKASI PENERJEMAH BAHASA ISYARAT) SEBAGAI SOLUSI DISEKUILIBRIUM KOMUNIKASI DAN PENDIDIKAN DI KOTA SEMARANG," *Journal For Energetic Youngsters*, vol. 2, no. 1, pp. 27-45, 2024.
- [4] M. B. S. Bakti, "PENGENALAN ANGKA SISTEM ISYARAT BAHASA INDONESIA DENGAN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK," *Seminar Nasional Inovasi Teknologi UN PGRI Kediri*, pp. 11-16, 2019.
- [5] I. J. Thira, "PENGENALAN ALFABET SISTEM ISYARAT BAHASA INDONESIA (SIBI) MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *Jurnal Algoritma*, vol. 20, no. 2, pp. 421-432, 2023.
- [6] F. A. B. J. Joelyan Vicky, "IMPLEMENTASI PENDETEKSI PENYAKIT PADA DAUN ALPUKAT MENGGUNAKAN METODE CNN," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, vol. 2, pp. 155-162, 2023.
- [7] N. L. M. Zahrah Fadhilah, "PENGENALAN ALFABET SIBI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK SEBAGAI MEDIA PEMBELAJARAN BAGI MASYARAKAT UMUM," *Jurnal*

Informatika: Jurnal pengembangan IT (JPIT), vol. 8, no. 2, pp. 162-168, 2023

- [8] R. H. Alfikri, "PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID," *JURNAL TEKNOINFO*, vol. 16, no. 2, pp. 183-197, 2022.
- [9] S. T. G. K. Marry Christin Assa, "INTERACTIVE APPLICATION TO LEARN INDONESIAN SIGN LANGUAGE," *Jurnal Teknik Elektro dan Komputer*, vol. 10, no. 2, pp. 135-144, 2021.
- [10] F. N. F. P. K. S. Gulpi Qorik Oktagalu Pratamasunu, "DETEKSI TANGAN OTOMATIS PADA VIDEO PERCAKAPAN BAHASA ISYARAT INDONESIA MENGGUNAKAN METODE DEEP GATED RECURRENT UNIT (GRU)," *Jurnal Politeknik Caltex Riau*, vol. 8, no. 1, p. 186 – 193 , 2022.
- [11] N. K. R. M. Irfan Maulana, "DETEKSI BENTUK WAJAH MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3348-3355, 2023.
- [12] I. D. U. Yunendah Nur Fuadah, "OPTIMASI CONVOLUTIONAL NEURAL NETWORK DAN K-FOLD CROSS VALIDATION PADA SISTEM KLASIFIKASI GLAUKOMA," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 3, pp. 728 - 741, 2022.
- [13] M. F. Indra Bakti, "ARSITEKTUR CONVOLUTIONAL NEURAL NETWORK INCEPTIONRESNET-V2 UNTUK PENGELOMPOKAN PNEUMONIA CHEST X-RAY," *Jurnal Komputer dan Teknologi (JUKOMTEK)* , vol. 1, no. 2, p. 35–42, 2023.

- [14] A. Kholik, "KLASIFIKASI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA TANGKAPAN LAYAR HALAMAN INSTAGRAM," *JDMSI*, vol. 2, no. 2, pp. 10-20, 2021.
- [15] H. B. H. ., Y. Y. Naim Rochmawati, "ANALISA LEARNING RATE DAN BATCH SIZE PADA KLASIFIKASI COVID MENGGUNAKAN DEEP LEARNING DENGAN OPTIMIZER ADAM," *Journal Information Engineering and Educational Technology*, vol. 5, no. 2, pp. 44-48, 2021.
- [16] J. J. S. Yeremia Yosefan Pane, "KLASIFIKASI JENIS BURUNG MENGGUNAKAN METODE TRANSFER LEARNING," *Jurnal Teknologi Terpadu*, vol. 9, no. 2, pp. 89-94, 2023.
- [17] E. Y. H. M. N. P. L. Ni Made Kirei Kharisma Handayani, "PENGENALAN EKSPRESI WAJAH MENGGUNAKAN TRANSFER LEARNING MOBILENETV2 DAN EFFICIENTNET-B0 DALAM MEMPREDIKSI PERKELAHIAN," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, pp. 106-114, 2024.
- [18] I. Rochmawati, "ANALISIS USER INTERFACE SITUS WEB IWEARUP.COM," vol. 7, no. 2, pp. 31-44, 2019.
- [19] M. A. I. P. Ronaldo Kristoforus Ngantung, "MODEL PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS USER-CENTERED DESIGN MENGGUNAKAN FRAMEWORK FLASK PYTHON," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, pp. 1051-1062, 2021.
- [20] T. R. Maulana Sidiq, "PERANCANGAN APLIKASI PENJUALAN BERBASIS WEB DENGAN METODE PROTOTYPING UNTUK UMKM SINAR TERANG DESA PUSAKASARI KECAMATAN CIPAKU," *NFOTECH journal*, vol. 9, no. 1, pp. 76-83, 2023.
- [21] A. F. S. A. M. W. Tia Arianti, "PERANCANGAN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN DIAGRAM UML (UNIFIED

- MODELLING LANGUAGE)," *Jurnal Ilmiah Komputer Terapan dan Informasi (JIKTI)*, vol. 1, no. 1, pp. 19-25, 2022.
- [22] L. Setiyani, "DESAIN SISTEM: USE CASE DIAGRAM," *Seminar Nasional : Inovasi & Adopsi Teknologi 2021*, pp. 246-260, 2021.
- [23] S. F. P. D. M. M Nazir, "PERANCANGAN APLIKASI E-VOTING MENGGUNAKAN DIAGRAM UML (UNIFIED MODELLING LANGUAGE)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 1, no. 1, pp. 5-9, 2022.
- [24] P. D. Arnesia, "APLIKASI ARTIFICIAL INTELLIGENCE UNTUK MENDETEKSI OBJEK BERBASIS WEB MENGGUNAKAN LIBRARY TENSORFLOW JS, REACT JS DAN COCO DATASET," *Jurnal Sistem Informasi*, vol. 9, no. 1, pp. 62-69, 2022.
- [25] I. A. Shaleh, "PENGUJIAN BLACK BOX PADA SISTEM INFORMASI PENJUALAN BUKU BERBASIS WEB DENGAN TEKNIK EQUIVALENT PARTITIONS," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 4, no. 1, pp. 38-45, 2021.
- [26] A. Aziz, "PERANCANGAN DAN IMPLEMENTASI APLIKASI PENJUALAN CNG INDONESIA PRODUCT BERBASIS ANDROID DENGAN METODE WHITE BOX," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 3, pp. 1861-1872, 2022.
- [27] I. Wahyudi, "ANALISIS BLACKBOX TESTING DAN USER ACCEPTANCE TESTING TERHADAP SISTEM INFORMASI SOLUSIMEDSOSKU," *Jurnal Teknosains Kodepena*, vol. 4, no. 1, pp. 1-9, 2023.
- [28] A. Z. Syahputri, "KERANGKA BERFIKIR PENELITIAN KUANTITATIF," *Jurnal Ilmu Pendidikan dan Pengajaran*, vol. 2, no. 1, pp. 160-166, 2023.
- [29] T. D. K. P. I. A. Mochammad Toyib, "PENERAPAN ALGORITMA CNN UNTUK MENDETEKSI TULISAN TANGAN ANGKA ROMAWI

DENGAN AUGMENTASI DATA," *Algoritma : Jurnal Matematika, Ilmu pengetahuan Alam, Kebumihan dan Angkasa*, vol. 2, no. 3, pp. 108-120, 2024.

LAMPIRAN

Lampiran 1 Lembar Bimbingan Dosen Pembimbing 1



LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Khoir Afandi
 NPM : 20670121
 Program Studi : Informatika
 Judul Skripsi : APLIKASI PENGENALAN HURUF ABJAD
 BAHASA ISYARAT MENGGUNAKAN METODE
 CONVOLUTIONAL NEURAL NETWORK (CNN)
 BERBASIS REACT JS
 Dosen Pembimbing I : Aris Tri Jaka Harjanta, S.Kom., M.Kom.
 Dosen Pembimbing II : Febrian Murti Dewanto, SE., M. Kom.

No.	Hari Tanggal	Uraian Bimbingan	Paraf
		Induksi	<i>[Signature]</i>
		Konsep 1	<i>[Signature]</i>
		Pertemuan lanjutan form + CNN	<i>[Signature]</i>
		Rev. kerangka / kata kunci	<i>[Signature]</i>
		Cari referensi / user dan eye	<i>[Signature]</i>
		BAB II dan	<i>[Signature]</i>
		BAB III Lanjutan	<i>[Signature]</i>
		buat kerangka gambar di BAB IV	<i>[Signature]</i>

Dosen Pembimbing I,
[Signature]
 Aris Tri Jaka Harjanta, S.Kom., M.Kom.
 NIDN. 0619048202

Mahasiswa,
[Signature]
 Khoir Afandi
 NPM. 20670121



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidosodi Timur Nomor 24 Dr. Cipto, Semarang - Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrisng@gmail.com, Homepage : www.upgrisng.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Khoir Afandi
 NPM : 20670121
 Program Studi : Informatika
 Judul Skripsi : APLIKASI PENGENALAN HURUF ABJAD BAHASA
 ISYARAT MENGGUNAKAN METODE
 CONVOLUTIONAL NEURAL NETWORK (CNN)
 BERBASIS REACT JS
 Dosen Pembimbing I : Aris Tri Jaka Harjanta, S.Kom., M.Kom.
 Dosen Pembimbing II : Febrian Murti Dewanto, SE., M. Kom.

No.	Hari Tanggal	Uraian Bimbingan	Paraf
		HB IV de	2/1
		BAB V Ace	3/1
		Ace Daftar ordg.	3/1

Dosen Pembimbing I,

 Aris Tri Jaka Harjanta, S.Kom., M.Kom.
 NIDN. 0619048202

Mahasiswa,

 Khoir Afandi
 NPM. 20670121

Lampiran 2 Lembar Bimbingan Dosen Pembimbing 2



UNIVERSITAS PGRI SEMARANG
FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Khoir Afandi
 NPM : 20670121
 Program Studi : Informatika
 Judul Skripsi : APLIKASI PENGENALAN HURUF ABJAD
 BAHASA ISYARAT MENGGUNAKAN METODE
 CONVOLUTIONAL NEURAL NETWORK (CNN)
 BERBASIS REACT JS
 Dosen Pembimbing I : Aris Tri Jaka Harjanta, S.Kom.,M.Kom.
 Dosen Pembimbing II : Febrian Murti Dewanto, SE., M. Kom.

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1		ACC judul	AT
2	26/5	Revisi proposal.	AT
3	30/5	Revisi proposal bab 3	AT
4	11/6	ACC proposal.	AT
5	10/6	Revisi bab 4.	AT
6		Revisi bab 4	AT
7		ACC Bab 4	AT
8	4/7	ACC Skripsi	AT

Dosen Pembimbing II,

 Febrian Murti Dewanto, SE., M. Kom.
 NIDN. 0626028201

Mahasiswa,

 Khoir Afandi
 NPM. 20670121

Lampiran 3 Kuisiener Pengujian Black Box

**KUESIONER PENGUJIAN *BLACK BOX* PADA APLIKASI PENGENALAN
HURUF ABJAD BAHASA ISYARAT MENUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS REACT JS**

Nama Penguji : Bambang Agus H, S, tam, M, tom
Tanggal Pengujian : 20 Juni 2024

Nama pengujian	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	keterangan	
				diterima	ditolak
Halaman home	User dapat melakukan gulir atau scroll kebawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi (upload), form input teks ke bahasa isyarat, dan daftar huruf Bahasa isyarat bisindo)	Website akan menampilkan semua fitur yang ada pada halaman home	✓	
	User dapat menekan tombol realtime page untuk berganti ke page realtime	User dapat melihat bagian realtime page untuk prediksi secara realtime	Website menampilkan page bagian realtime untuk prediksi	✓	
Halaman home (prediksi upload)	User dapat menekan tombol choose file	User dapat memilih gambar yang akan di upload untuk di prediksi dari file manager	Website akan menampilkan file manager, menampilkan preview dari gambar yang akan di prediksi, dan	✓	

			akan menampilkan hasil prediksi		
Halaman home (prediksi realtime)	User dapat mengarahkan objek ke kamera untuk di prediksi	User dapat melihat hasil prediksi realtime	Website menampilkan hasil prediksi realtime	✓	
	User dapat menekan tombol on/of camera	User dapat mengaktifkan dan menonaktifkan kamera	Website dapat menampilkan hasil ketika kamera aktif dan non aktif	✓	
Halaman home (teks ke Bahasa isyarat)	User dapat melakukan terjemah dari teks ke Bahasa isyarat	User dapat memasukan keyword teks yang akan di terjemah ke bahasa isyarat	Website dapat memunculkan terjemahan hasil	✓	

Saran dari penguji :

.....

.....

.....

Semarang,


 Semarang, 21/05/2023 (Siti Nurrahma)

**KUESIONER PENGUJIAN *BLACK BOX* PADA APLIKASI PENGENALAN
HURUF ABJAD BAHASA ISYARAT MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS REACT JS**

Nama Penguji : NOORA Q.N.
Tanggal Pengujian : 20 Juni 2024.

Nama pengujian	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	keterangan	
				diterima	ditolak
Halaman home	User dapat melakukan gulir atau scroll kebawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi (upload), form input teks ke bahasa isyarat, dan daftar huruf Bahasa isyarat bisindo)	Website akan menampilkan semua fitur yang ada pada halaman home	✓	
	User dapat menekan tombol realtime page untuk berganti ke page realtime	User dapat melihat bagian realtime page untuk prediksi secara realtime	Website menampilkan bagian realtime untuk prediksi	✓	
Halaman home (prediksi upload)	User dapat menekan tombol choose file	User dapat memilih gambar yang akan di upload untuk di prediksi dari file manager	Website akan menampilkan file manager, menampilkan preview dari gambar yang akan di prediksi, dan	✓	

			akan menampilkan hasil prediksi		
Halaman home (prediksi realtime)	User dapat mengarahkan objek ke kamera untuk di prediksi	User dapat melihat hasil prediksi realtime	Website menampilkan hasil prediksi realtime	✓	
	User dapat menekan tombol on/of camera	User dapat mengaktifkan dan menonaktifkan kamera	Website dapat menampilkan hasil ketika kamera aktif dan non aktif	✓	
Halaman home (teks ke Bahasa isyarat)	User dapat melakukan terjemah dari teks ke Bahasa isyarat	User dapat memasukan keyword teks yang akan di terjemah ke bahasa isyarat	Website dapat memunculkan terjemahan hasil	✓	

Saran dari penguji :

Hasil untuk kalimat masih berganti :
 bisa di atur dgn. memunculkan % prediksi
 huruf yg terdteksi ex. w so e. est.
 1 10%
 v 70%

Semarang,


 Nama

**KUESIONER PENGUJIAN *BLACK BOX* PADA APLIKASI PENGENALAN
HURUF ABJAD BAHASA ISYARAT MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS REACT JS**

Nama Penguji : Ramadhan Rezaldy, S.kom, M.kom

Tanggal Pengujian : 20 Juni 2024

Nama pengujian	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	keterangan	
				diterima	ditolak
Halaman home	User dapat melakukan gulir atau scroll kebawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi (upload), form input teks ke bahsa isyarat, dan daftar huruf Bahasa isyarat bisindo)	Website akan menampilkan semua fitur yang ada pada halaman home	✓	
	User dapat menekan tombol realtime page untuk berganti ke page realtime	User dapat melihat bagian realtime page untuk prediksi secara realtime	Website menampilkan page bagian realtime untuk prediksi	✓	
Halaman home (prediksi upload)	User dapat menekan tombol choose file	User dapat memilih gambar yang akan di upload untuk di prediksi dari file manager	Website akan menampilkan file manager, menampilkan preview dari gambar yang akan di prediksi, dan	✓	

			akan menampilkan hasil prediksi		
Halaman home (prediksi realtime)	User dapat mengarahkan objek ke kamera untuk di prediksi	User dapat melihat hasil prediksi realtime	Website menampilkan hasil prediksi realtime	✓	
	User dapat menekan tombol on/of camera	User dapat mengaktifkan dan menonaktifkan kamera	Website dapat menampilkan hasil ketika kamera aktif dan non aktif	✓	
Halaman home (teks ke Bahasa isyarat)	User dapat melakukan terjemah dari teks ke Bahasa isyarat	User dapat memasukan keyword teks yang akan di terjemah ke bahasa isyarat	Website dapat memunculkan terjemahan hasil	✓	

Saran dari penguji : Peletakkan di website hanya kurang 1000

.....

Semarang, 20 Juni 2019



Komachen Kresky

Lampiran 4 Pengujian Use Acceptance Testing

Kuesioner Pengujian User Acceptance Testing (UAT) pada Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Menggunakan Metode Convolutional Neural Network (CNN) Berbasis React JS

Nama Penguji : Ena Septiana Kristyanik
 Tanggal Pengujian : 21-06-2024

No	Pertanyaan	Skor				
		Tidak setuju	Kurang setuju	Cukup setuju	Setuju	Sangat setuju
Aspek kegunaan						
1.	Apakah aplikasi berguna bagi orang yang ingin belajar huruf abjad bahasa isyarat BISINDO?					✓
2.	Apakah aplikasi pengenalan bahasa isyarat memberikan informasi huruf abjad bahasa isyarat BISINDO?					✓
3.	Apakah aplikasi pengenalan bahasa isyarat dapat memprediksi huruf abjad bahasa isyarat BISINDO?					✓
Aspek kemudahan pengguna (user)						
4.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO mudah dioperasikan?					✓
5.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO					✓

	sesuai yang diharapkan?					
6.	Apakah Anda merasa ada fitur yang kurang atau perlu ditambahkan pada aplikasi ini?			✓		
Aspek tampilan (user interface)						
7.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO perlu dikembangkan lagi?				✓	

Keterangan :

- 1 = Tidak setuju
- 2 = Kurang setuju
- 3 = Cukup setuju
- 4 = Setuju
- 5 = Sangat setuju

.....
Jeni

.....
Ena Septama Istiyanti

Kuesioner Pengujian User Acceptance Testing (UAT) pada Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Menggunakan Metode Convolutional Neural Network (CNN) Berbasis React JS

Nama Penguji : *Sarman Tubagus Alfarizi*
 Tanggal Pengujian : *27 Juni 2021*

No	Pertanyaan	Skor				
		Tidak setuju	Kurang setuju	Cukup setuju	Setuju	Sangat setuju
Aspek kegunaan						
1.	Apakah aplikasi berguna bagi orang yang ingin belajar huruf abjad bahasa isyarat BISINDO?					✓
2.	Apakah aplikasi pengenalan bahasa isyarat memberikan informasi huruf abjad bahasa isyarat BISINDO?					✓
3.	Apakah aplikasi pengenalan bahasa isyarat dapat memprediksi huruf abjad bahasa isyarat BISINDO?					✓
Aspek kemudahan pengguna (user)						
4.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO mudah di operasikan?				✓	
5.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO				✓	

	sesuai yang diharapkan?					
6.	Apakah Anda merasa ada fitur yang kurang atau perlu ditambahkan pada aplikasi ini?					✓
Aspek tampilan (<i>user interface</i>)						
7.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO perlu dikembangkan lagi?					✓

Keterangan :

- 1 = Tidak setuju**
- 2 = Kurang setuju**
- 3 = Cukup setuju**
- 4 = Setuju**
- 5 = Sangat setuju**

.....

.....
Simon Tubagus A

Kuesioner Pengujian User Acceptance Testing (UAT) pada Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Menggunakan Metode Convolutional Neural Network (CNN) Berbasis React JS

Nama Penguji : **LIHAM LUTFI ADITYA** .
 Tanggal Pengujian : **02-06-2024** .

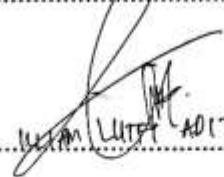
No	Pertanyaan	Skor				
		Tidak setuju	Kurang setuju	Cukup setuju	Setuju	Sangat setuju
Aspek kegunaan						
1.	Apakah aplikasi berguna bagi orang yang ingin belajar huruf abjad bahasa isyarat BISINDO?					✓
2.	Apakah aplikasi pengenalan bahasa isyarat memberikan informasi huruf abjad bahasa isyarat BISINDO?					✓
3.	Apakah aplikasi pengenalan bahasa isyarat dapat memprediksi huruf abjad bahasa isyarat BISINDO?					✓
Aspek kemudahan pengguna (user)						
4.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO mudah di operasikan?				✓	
5.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO					✓

	sesuai yang diharapkan?					
6.	Apakah Anda merasa ada fitur yang kurang atau perlu ditambahkan pada aplikasi ini?				✓	
Aspek tampilan (user interface)						
7.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang mudah dipahami?				✓	
8.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tema warna yang enak dilihat?			✓		
10.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO perlu dikembangkan lagi?				✓	

Keterangan :

- 1 = Tidak setuju
- 2 = Kurang setuju
- 3 = Cukup setuju
- 4 = Setuju
- 5 = Sangat setuju

Semarang, 22 Juni 2024.


.....
WULAN LUTFI ADITJA

Kuesioner Pengujian User Acceptance Testing (UAT) pada Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Mengunaka Metode Convulotional Neural Network (CNN) Berbasis React JS

Nama Penguji : RUSHAYATI
 Tanggal Pengujian : 21. JUNI 2024.

No	Pertanyaan	Skor				
		Tidak setuju	Kurang setuju	Cukup setuju	Setuju	Sangat setuju
Aspek kegunaan						
1.	Apakah aplikasi berguna bagi orang yang ingin belajar huruf abjad bahasa isyarat BISINDO?					✓
2.	Apakah aplikasi pengenalan bahasa isyarat memberikan informasi huruf abjad bahasa isyarat BISINDO?					✓
3.	Apakah aplikasi pengenalan bahasa isyarat dapat memprediksi huruf abjad bahasa isyarat BISINDO?					✓
Aspek kemudahan pengguna (user)						
4.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO mudah di operasikan?					✓
5.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO					

	sesuai yang diharapkan?					✓
6.	Apakah Anda merasa ada fitur yang kurang atau perlu ditambahkan pada aplikasi ini?					✓
Aspek tampilan (user interface)						
7.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tema warna yang enak dilihat?					✓
10.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO perlu dikembangkan lagi?					✓

Keterangan :

- 1 = Tidak setuju
- 2 = Kurang setuju
- 3 = Cukup setuju
- 4 = Setuju
- 5 = Sangat setuju

SEMARANG, 21 JUNI 2024



RUSHAYATI

Kuesioner Pengujian User Acceptance Testing (UAT) pada Aplikasi Pengenalan Huruf Abjad Bahasa Isyarat Menggunakan Metode Convolutional Neural Network (CNN) Berbasis React JS

Nama Penguji : PUSUK MANGGEL RAHAYU

Tanggal Pengujian : 21 JUNI 2024

No	Pertanyaan	Skor				
		Tidak setuju	Kurang setuju	Cukup setuju	Setuju	Sangat setuju
Aspek kegunaan						
1.	Apakah aplikasi berguna bagi orang yang ingin belajar huruf abjad bahasa isyarat BISINDO?					✓
2.	Apakah aplikasi pengenalan bahasa isyarat memberikan informasi huruf abjad bahasa isyarat BISINDO?					✓
3.	Apakah aplikasi pengenalan bahasa isyarat dapat memprediksi huruf abjad bahasa isyarat BISINDO?					✓
Aspek kemudahan pengguna (user)						
4.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO mudah di operasikan?					✓
5.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO					✓

	sesuai yang diharapkan?					
6.	Apakah Anda merasa ada fitur yang kurang atau perlu ditambahkan pada aplikasi ini?					✓
Aspek tampilan (<i>user interface</i>)						
7.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO memiliki tema warna yang enak dilihat?					✓
10.	Apakah aplikasi pengenalan huruf abjad bahasa isyarat BISINDO perlu dikembangkan lagi?					✓

Keterangan :

- 1 = Tidak setuju
- 2 = Kurang setuju
- 3 = Cukup setuju
- 4 = Setuju
- 5 = Sangat setuju

SEMARANG, 21 JUNI 2024



PUPIT MANBEE RAHAYU

Lampiran 5 Lembar Revisi Ujian

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Khoir Afandi
 NPM : 20670121
 Judul : APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
 MENGGUNAKAN METODE CONVOLUTIONAL NEURAL
 NETWORK (CNN) BERBASIS REACT JS

No	Uraian Revisi	Keterangan
1	Tangkapan prototyping belum ditentukan secara Default pada pembahasannya.	
2	Use case diagram menyajikan sebuah aplikasi & bahasa sumber data.	
3	pada pembahasannya Alur penode data mulai dari penyimpulan data, training data dll' belum ditentukan secara detail. (dapat ditambahkan gambar)	
4	Detail Library JS digunakan?	
5	Dokumentasi pada saat uji coba.	
6	React JS belum dibahas pada pembahasannya.	
7	Epoch Accuracy 96% why?	

Pengesahan Penguji III



Bambang Agus H. S. Kom, M. Kom
 NIP/NPP. 148201433

*1) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Khoir Afandi
 N P M : 20670121
 Judul : APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
 MENUNAKAN METODE CONVOLUTIONAL NEURAL
 NETWORK (CNN) BERBASIS REACT JS

No	Uraian Revisi	Keterangan
1.	Perbaikan Abstract	
2.	Ubah kata pada pendahuluan	
3.	Ubah & lengkapi kepingan	

Pengesahan Penguji I

Aris Tri Joko Harjanto S.Kom., M.Kom
 NIP/NPP. 148201443

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Khok Alandi

N P M : 20670121

Judul : APLIKASI PENGENALAN HURUF ABJAD BAHASA ISYARAT
MENGUNAKAN METODE CONVOLUTIONAL NEURAL
NETWORK (CNN) BERBASIS REACT JS

No	Uraian Revisi	Keterangan
1.	Pop her p... ..	
2.	Form aplikasi di p... ..	
3.	lampiran (gambar ?).	
4.	Yang di lipat tyro	
	<p>Penjelasan</p> <p>CNN → transfer learning.</p> <p>Jelaskan (karena .7, 3 t.</p> <p>layer yg di transfer.</p> <p>$R = \frac{1}{2}$</p> <p>S =</p>	

Pengesahan Penguji II



Febrian Murti Dewanto, SE, M. Kom

NIP/NPP, 057801172

*1 Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

Lampiran 6 Dokumentasi UAT



