

**PENERAPAN ALGORITMA *GAUSSIAN BLUR* DALAM SISTEM
PENERJEMAH BAHASA ISYARAT**

SKRIPSI

FACHRIZA ADRIAN

201401099



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

**PENERAPAN ALGORITMA *GAUSSIAN BLUR* DALAM SISTEM
PENERJEMAH BAHASA ISYARAT**

SKRIPSI

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer**

FACHRIZA ADRIAN

201401099



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2024

PERSETUJUAN

Judul : PENERAPAN ALGORITMA *GAUSSIAN BLUR*
DALAM SISTEM PENERJEMAH BAHASA
ISYARAT
Kategori : SKRIPSI
Nama : FACHRIZA ADRIAN
Nomor Induk Mahasiswa : 201401099
Program Studi : SARJANA (S-1) ILMU KOMPUTER
Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Pembimbing 2



Dewi Sartika Br. Ginting, S.Kom., M.Kom.
NIP. 199005042019032023

Pembimbing 1

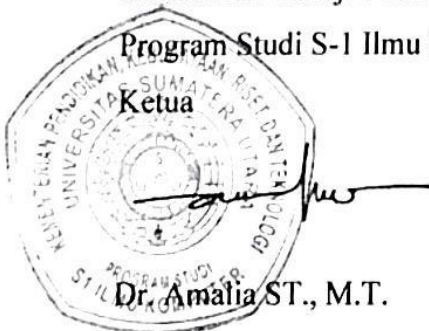


Dr. Eng Ade Candra, ST., M.Kom.
NIP. 197909042009121002

Diketahui/Disetujui Oleh

Program Studi S-1 Ilmu Komputer

Ketua



Dr. Amalia ST., M.T.

NIP. 197812212014042001

PERNYATAAN**PENERAPAN ALGORITMA *GAUSSIAN BLUR* DALAM SISTEM PENERJEMAH
BAHASA ISYARAT****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil penelitian saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah dicantumkan sumbernya.

Medan, 12 Juni 2024



Fachriza Adrian

201401099

PENGHARGAAN

Dengan menyebut nama Allah Yang Maha Pengasih dan Maha Penyayang, penulis menyampaikan terima kasih yang tak terbatas atas nikmat dan bimbingan-Nya yang telah memfasilitasi penyelesaian skripsi ini yang merupakan syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi S-1 Ilmu Komputer di Universitas Sumatera Utara. Penulis pun menghaturkan doa serta salam kepada Nabi Muhammad *Shalallaahu 'Alayhi Wasallam* yang telah membimbing umat ke zaman pencerahan saat ini.

Penulis ingin menyampaikan terimakasih kepada ibu, Hj. Dini Handayani Harahap, untuk semua dedikasi, kasih, dan perlindungan yang luar biasa serta doa yang senantiasa mendampingi dalam perjalanan ini. Apresiasi tak terhingga juga untuk ayah, H. Arpiansyah SE. atas kasih sayang yang selalu memberikan penulis kekuatan dalam setiap langkah yang ditempuh. Terima kasih juga disampaikan kepada semua pihak yang telah membantu dan mendukung hingga tiba pada tahap ini.

Skripsi ini merupakan hasil dari kontribusi dan dukungan dari berbagai individu untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Prof. Dr. Muryanto Amin S.Sos., M.Si., Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., Dekan Fakultas Ilmu Komputer dan Teknologi Informasi di Universitas Sumatera Utara.
3. Ibu Dr. Amalia, S.T., M.T., sebagai Ketua Program Studi S-1 Ilmu Komputer yang juga membimbing penulis sebagai Dosen Penguji I dan Pembimbing Akademik dengan berbagai masukan dan kritik yang konstruktif selama penyusunan skripsi ini.
4. Bapak Dr. Eng Ade Candra, ST., M.Kom., Dosen Pembimbing I yang telah memberikan bimbingan dan masukan yang sangat berharga selama penulisan skripsi.
5. Ibu Dewi Sartika Br.Ginting, S.Kom., M.Kom., Dosen Pembimbing II yang memberikan nasihat dan kritik yang membantu dalam penyusunan skripsi ini.
6. Bapak Ivan Jaya S.Si., M.Kom., Dosen Penguji II yang telah memberikan saran dan kritik yang konstruktif dalam penyelesaian skripsi ini.

7. Semua dosen di Fasilkom-TI USU, khususnya dari Program Studi S-1 Ilmu Komputer, yang telah memberikan edukasi dan nilai-nilai moral yang sangat berharga.
8. Seluruh civitas akademik Fasilkom-TI USU yang telah memberikan dukungan selama masa studi.

Penulis juga berterima kasih kepada semua yang telah mendukung dan mendoakan penulis yang tidak mungkin disebutkan satu per satu. Semoga Allah *Subhanahu Wa Ta'ala* senantiasa memberikan keberkahan dan kebaikan atas segala dukungan yang diberikan dan semoga penelitian ini dapat memberikan manfaat dan menginspirasi untuk masa depan.

Medan, 12 Juni 2024

Penulis,



Fachriza Adrian

ABSTRAK

Pada tahun 2021, diperkirakan sekitar 1,3 miliar orang, atau sekitar 16% dari populasi dunia mengalami berbagai jenis kecacatan termasuk disabilitas sensorik seperti tunarungu dan tunawicara. Meski bahasa isyarat digunakan sebagai alat komunikasi non-verbal, gerakannya sering kali sulit dipahami oleh orang yang tidak terbiasa dengan bahasa isyarat. Layanan penerjemah bahasa isyarat telah meningkat dalam beberapa tahun terakhir tetapi masih memiliki keterbatasan. Teknologi *Deep Learning*, khususnya *Convolutional Neural Network* (CNN) telah menunjukkan kemajuan dalam pengenalan gambar dan video. CNN merupakan arsitektur yang efektif untuk klasifikasi objek dan telah digunakan untuk meningkatkan deteksi perubahan gerakan dalam pengenalan gerakan tangan. Teknik *Gaussian Blur* digunakan untuk mengurangi gangguan visual pada gambar sehingga dapat meningkatkan fokus pada pengenalan gerakan tangan dalam bahasa isyarat. Penelitian ini mengembangkan sistem penerjemah bahasa isyarat yang menerapkan blur latar belakang. Library *Tensorflow.js* dan *BodyPix* dapat dimanfaatkan untuk menerapkan teknik ini dengan memisahkan bagian tubuh dan latar belakang sehingga blur hanya diterapkan pada latar belakang tanpa mengganggu bagian tubuh. Dataset yang digunakan dalam penelitian ini terdiri dari 3.480 gambar gerakan bahasa isyarat, 2.420 gambar untuk data latih dan 1.040 untuk data validasi dengan 20 kategori/kelas. Model yang digunakan adalah *Convolutional Neural Network* yang dilatih menggunakan dataset tersebut hingga mencapai akurasi model sebesar 91%. Model ini kemudian digunakan untuk mengklasifikasikan gerakan bahasa isyarat menjadi kata atau kalimat.

Kata Kunci: *Sign language, Gesture recognition, Gaussian Blur, Deep Learning, Convolutional Neural Network*

ABSTRACT***APPLICATION OF THE GAUSSIAN BLUR ALGORITHM IN A SIGN LANGUAGE
TRANSLATOR SYSTEM***

In 2021, it was estimated that around 1.3 billion people or approximately 16% of the world's population experienced various types of disabilities including sensory disabilities such as deafness and muteness. Although sign language is used as a non-verbal communication tool. People with unfamiliar Sign language movements are often difficult to understand it. Sign language interpreter services have increased in recent years but still have limitations. Deep Learning technology, particularly Convolutional Neural Networks (CNN) has shown advancements in image and video recognition. CNN is an effective architecture for object classification and has been used to enhance motion detection in hand gesture recognition. The Gaussian Blur technique is used to reduce visual noise in images thereby improving focus on hand gesture recognition in sign language. This research develops a sign language interpreter system that applies background blur. The Tensorflow.js and BodyPix libraries can be utilized to implement this technique by separating body parts from the background allowing the blur to be applied to the background without affecting the body parts. The dataset used in this research consists of 3,480 images of sign language movements, 2,420 images for training data and 1,040 for validation data with 20 categories/classes. The model used is Convolutional Neural Network which was trained using this dataset achieve model accuracy of 91%. This model used to classify sign language movements into words or sentences.

Keywords: *Sign language, Gesture recognition, Gaussian Blur, Deep Learning, Convolutional Neural Network*

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN.....	iii
PENGHARGAAN.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Penelitian Relevan	5
1.8 Sistematika Penulisan	6
BAB 2	8
LANDASAN TEORI	8
2.1 Bahasa Isyarat	8
2.1.1 <i>Tunarungu</i>	8
2.1.2 <i>Tunawicara</i>	8
2.2 Sistem Isyarat Bahasa Indonesia	9
2.3 Artificial Intelligence.....	9
2.4 Deep Learning	9
2.5 Machine Learning	10
2.6 Convolutional Neural Network	10
2.7 Computer Vision	12
2.8 Algoritma Gaussian Blur	12
BAB 3	15
ANALISIS DAN PERANCANGAN.....	15

3.1	Analisis	15
3.1.1	<i>Analisis Masalah.....</i>	15
3.1.2	<i>Analisis Kebutuhan</i>	15
3.2	Perancangan Sistem	16
3.2.1	Diagram Umum Sistem.....	17
3.2.2	<i>Use Case Diagram</i>	18
3.2.3	<i>Activity Diagram</i>	19
3.3	Flowchart (Diagram Alir)	20
3.3.1	<i>Flowchart Background blur.....</i>	21
3.3.2	<i>Flowchart Sistem Penerjemah Bahasa Isyarat.....</i>	22
3.4	Analisis Dataset	22
3.5	Perancangan Pre-processing Dataset	23
3.5.1	<i>Data Resizing</i>	23
3.5.2	<i>Image Annotation</i>	23
3.5.3	<i>Data Augmentasi.....</i>	24
3.6	Perancangan Teknik <i>Background Blur</i>	25
3.6.1	<i>Tensorflow.js.....</i>	25
3.6.2	<i>BodyPix</i>	25
3.7	Perancangan Interface Website	27
BAB 4	28
IMPLEMENTASI DAN PENGUJIAN	28
4.1	Implementasi Sistem	28
4.1.1	Dataset.....	28
4.2	Implementasi Interface Website	28
4.2.1	Halaman Home	28
4.2.2	Halaman Penerjemah	29
4.3	Implementasi Teknik <i>Background Blur</i>	29
4.3.1	<i>Tensorflow.js.....</i>	29
4.3.2	<i>Bodypix</i>	31
4.3.3	Segmentasi dan Efek Blur.....	31
4.4	Pengujian Sistem	32
4.4.1	Pre-Processing Data	32
4.4.2	Pengujian Model	36
4.4.3	Hasil Pengujian	39

BAB 5	49
PENUTUP.....	49
5.1 Kesimpulan	49
5.2 Saran.....	49
DAFTAR PUSTAKA.....	50

DAFTAR TABEL

Tabel 4. 1 Kosakata dan anotasi.....	42
Tabel 4. 2 List Kalimat.....	45

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Convolutional Neural Network	11
Gambar 2.2 Grafik Fungsi Gaussian	14
Gambar 3.1 Gambaran Umum Sistem	17
Gambar 3.2 Rancangan Sistem.....	18
Gambar 3.3 Use Case Diagram	19
Gambar 3.4 Activity Diagram	20
Gambar 3.5 Flowchart Background blur	21
Gambar 3.6 Flowchart Sistem Penerjemah Bahasa Isyarat.....	22
Gambar 4.1 Halaman Home	28
Gambar 4.2 Halaman Penerjemah.....	29
Gambar 4.3 Import Library Tensorflow.js	29
Gambar 4.4 setupCamera	30
Gambar 4.5 startSegmentation	31
Gambar 4.6 updateSegmentation	31
Gambar 4.7 Segmentasi dan Efek Blur	31
Gambar 4.8 <i>Data Resizing</i>	32
Gambar 4.9 Modul <i>Mediapipe</i>	33
Gambar 4.10 Proses Anotasi Gambar Dataset	34
Gambar 4.11 Proses Augmentasi	35
Gambar 4.12 <i>Training Model</i>	36
Gambar 4.13 <i>Epoch Accuracy</i>	37
Gambar 4.14 <i>Epoch Loss</i>	38
Gambar 4.15 <i>Confusion Matrix</i>	39
Gambar 4.16 Halaman Home	40
Gambar 4.17 Halaman Penerjemah Kata	40
Gambar 4.18 <i>Background Blur</i>	41
Gambar 4.19 "Saya Senang"	46
Gambar 4.20 "Hai Ibu"	46
Gambar 4.21 "Pesawat Terbang"	46
Gambar 4.22 "Kita Sedih".....	47
Gambar 4.23 "Maaf Ibu".....	47

Gambar 4.24	"Ayah Sakit"	48
Gambar 4.25	"Diam Mulut Kamu"	48

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Diperkirakan sekitar 1,3 miliar orang atau 16% dari total populasi dunia mengalami kecacatan yang signifikan pada tahun 2021. Cacatan tersebut melibatkan berbagai kondisi termasuk cacat fisik, sensorik, mental, atau perkembangan. Beragam kondisi ini menggambarkan kompleksitas tantangan yang dihadapi oleh kelompok difabel dalam kehidupannya (WHO, 2021).

Tunarungu dan tunawicara merupakan kelompok difabel yang mengalami gangguan sensorik. Mereka menggunakan bahasa isyarat sebagai bentuk komunikasi non-verbal untuk menyampaikan emosi dan pemikiran mereka kepada orang lain. Namun, bagi individu yang tidak familiar dalam memahami gerakan bahasa isyarat hal ini bisa menjadi sulit sehingga bantuan pelatihan dalam penguasaan bahasa isyarat diperlukan terutama dalam konteks janji medis, hukum, serta pendidikan (Sabeenian et al., 2020).

Dalam beberapa tahun permintaan terhadap layanan penerjemah bahasa isyarat telah meningkat. Layanan alternatif seperti Video Remote Interpreting yang menggunakan koneksi internet memberikan kemudahan bagi pengguna layanan penerjemah bahasa isyarat. Meskipun demikian layanan ini tetap memiliki beberapa keterbatasan (Sabeenian et al., 2020).

Sistem penerjemah bahasa isyarat bertujuan untuk memungkinkan tunarungu dan tunawicara berkomunikasi dengan tepat (Shah et al., 2021). Klasifikasi berbasis gambar, video, dan tingkat kalimat adalah beberapa permasalahan dari sistem penerjemah bahasa isyarat (Suharjito et al., 2021). Terkait dengan permasalahan tersebut, teknologi Deep Learning dapat menjadi salah satu solusi.

Deep Learning telah mencapai tingkat kemajuan yang lebih tinggi dalam berbagai domain dan aplikasi. Meskipun demikian tantangan signifikan masih dihadapi dalam pengenalan, penerjemahan, dan pembuatan video bahasa isyarat

dari perspektif pengembangan. Akurasi dalam pengenalan dan kualitas visual masih menjadi isu yang perlu diperhatikan (Natarajan et al., 2022).

Convolutional Neural Network (CNN) adalah salah satu model Deep Learning yang dapat digunakan untuk mengidentifikasi dan mengklasifikasikan objek secara otomatis dalam pengenalan gambar dan video. Deep learning dan Convolutional Neural Network (CNN) merupakan dua konsep yang erat kaitannya (Vakalopoulou et al., 2023).

Convolutional Neural Network (CNN) merupakan arsitektur yang efektif dalam mengklasifikasi objek karena mencapai akurasi prediksi yang baik dan mengonsumsi lebih sedikit sumber daya komputasi (Ce & Tie, 2020). Deteksi perubahan gerakan dalam pengenalan gerakan tangan telah ditingkatkan melalui penggunaan arsitektur Convolutional Neural Network (CNN), meskipun saat ini masih dilakukan secara statis (Adithya & Rajesh, 2020).

Dalam teknik pengenalan gerakan tangan, kamera digunakan untuk mengenali gerakan bahasa isyarat dan mengubahnya menjadi teks yang sesuai agar tunarungu dan tunawicara dapat memahaminya (Das et al., 2020). Pengenalan gerakan tangan secara real-time merupakan hal yang menantang karena beberapa hal. Pertama, gerakan tangan yang terlalu cepat dapat membuat gerakan menjadi kabur sehingga sulit terdeteksi oleh sistem. Kedua, sistem secara akurat mengenali gerakan yang serupa sehingga menghasilkan kesalahan dalam penerjemahan yang mengurangi akurasi deteksi. Ketiga, latar belakang atau background yang berantakan dan variasi pencahayaan sekitar juga menjadi faktor utama dalam pengenalan gerakan tangan (Huang et al., 2023).

Background Blur dalam gambar disebut sebagai teknik fotografi Panning. Foto diambil saat kamera sedang mengikuti pergerakan subjek yang bergerak sehingga menciptakan gambar subjek yang tajam dengan latar belakang yang buram akibat pergerakan relatif (Tabellion et al., 2023). Proses ini menghasilkan foto dimana satu objek yang ada dalam gambar muncul secara tajam sementara elemen di luar objek tersebut menampilkan efek blur. Efek visual ini menghasilkan fokus pada objek utama sementara elemen di sekitarnya menjadi blur. Kondisi ini memberikan dimensi dan penekanan visual pada bagian tertentu dari gambar (Ignatov et al., 2020).

Salah satu filter perataan gambar yaitu Gaussian Blur memanfaatkan fungsi Gaussian yang merujuk pada fungsi normal distribution untuk menghitung transformasi yang diterapkan pada setiap piksel dalam gambar (Shreya, 2021). Filter Gaussian Blur digunakan pada setiap frame untuk menghilangkan noise yang tidak diinginkan dan menghasilkan gambar dengan kedalaman yang bersih (Chandwani et al., 2023).

Berdasarkan uraian di awal maka penelitian ini mengimplementasikan teknik Background Blur untuk mengurangi gangguan visual pada gambar dan menghasilkan fokus pada pengenalan gerakan tangan dengan kedalaman gambar yang bersih agar akurasi prediksi meningkat. Dataset yang akan digunakan pada penelitian ini dikumpulkan langsung oleh peneliti.

1.2 Rumusan Masalah

Sistem penerjemah bahasa isyarat memungkinkan penyandang tunarungu dan tunawicara berkomunikasi dua arah dengan non difabel. Namun, dalam sistem penerjemah bahasa isyarat gerakan tangan sering kali tidak terdeteksi sehingga menghasilkan output yang kosong atau salah. Hal tersebut terjadi karena latar belakang atau *background* yang berantakan dan memiliki variasi pencahayaan yang berbeda sehingga kamera sulit mendeteksi area fokus dari pergerakan tangan. Oleh karena itu dibutuhkan implementasi *background blur* untuk memperbaiki kemampuan sistem penerjemah bahasa isyarat.

1.3 Batasan Masalah

Berikut adalah beberapa keterbatasan yang dihadapi dalam penelitian ini:

1. Penelitian ini difokuskan pada penerapan teknik *background blur* menggunakan algoritma *Gaussian Blur* khususnya dalam konteks Sistem Penerjemah Bahasa Isyarat.
2. Penelitian ini tidak membahas aspek pengembangan atau penerapan algoritma untuk pengenalan gerakan tangan, melainkan fokus pada perbaikan kualitas gambar dari gerakan tangan melalui teknik *background blur*.
3. Penerapan algoritma *Gaussian Blur* untuk mengoptimalkan kualitas gambar dari gerakan tangan, memperbaiki kontras antara objek utama dan

latar belakang serta mengurangi efek "halus" atau "berwarna" yang ada pada latar belakang dalam Sistem Penerjemah Bahasa Isyarat.

4. Penerapan teknik Background Blur ini hanya akan di terapkan pada webcam built in pada laptop.
5. Program akan dirancang berbasis Website.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan teknik *background blur* menggunakan algoritma *Gaussian Blur* dalam sistem penerjemah bahasa isyarat. Dengan menerapkan efek blur pada latar belakang, gerakan bahasa isyarat dapat lebih jelas teridentifikasi sehingga meningkatkan akurasi prediksi.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

- a. Memperbaiki kontras antara objek utama dan background gambar agar sistem lebih efektif dalam mengenali gerakan tangan dan menerjemahkan bahasa isyarat.
- b. Mengurangi noise atau gangguan yang mungkin terjadi dalam gambar atau video bahasa isyarat sehingga meningkatkan kualitas gambar yang diterjemahkan untuk dianalisis oleh sistem.
- c. Algoritma *Gaussian Blur* dapat mengoptimalkan kualitas gambar sehingga dapat mengurangi terjadinya overfitting yang dapat menyebabkan penurunan akurasi prediksi dalam sistem.

1.6 Metodologi Penelitian

Beberapa metode yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Studi Pustaka

Pada tahap ini penelitian diawali dengan mengumpulkan referensi dari berbagai sumber yang kredibel dan melakukan tinjauan literatur melalui buku-buku, jurnal, *e-book*, artikel ilmiah, makalah ataupun situs internet yang berhubungan dengan *Background blur*, *Gaussian Blur*, Sistem Penerjemah Bahasa Isyarat, dan *Convolutional Neural Network*.

2. Analisis dan Perancangan

Berdasarkan ruang lingkup penelitian, penulis melakukan analisis dan perancangan mengenai kebutuhan penelitian yang akan diwujudkan dalam bentuk diagram alir (*flowchart*).

3. Implementasi Sistem

Pada tahap ini, membuat sebuah sistem berbasis website yang dikembangkan dengan menggunakan bahasa pemrograman Python sebagai program dan HTML sebagai antarmuka sesuai dengan diagram alir yang telah dirancang.

4. Pengujian Sistem

Pada tahap ini, sistem yang telah dirancang dilakukan uji coba untuk melakukan penerjemah terhadap bahasa isyarat yang sudah menerapkan teknik *background blur*.

5. Dokumentasi Sistem

Pada tahap ini, penelitian yang telah dilakukan didokumentasikan mulai dari tahap analisis sampai kepada pengujian dalam bentuk skripsi.

1.7 Penelitian Relevan

Beberapa penelitian yang relevan dengan penelitian ini meliputi:

1. Penelitian berjudul "Penerapan Metode Deep Learning pada Aplikasi Pembelajaran Bahasa Isyarat Indonesia dengan Convolutional Neural Network (Studi Kasus: SLB-BC Mahardika Depok)" (Anggraini & Zakaria, 2023), telah berhasil mengembangkan aplikasi pembelajaran interaktif yang menggunakan Machine Learning dan Deep Learning melalui teknologi Convolutional Neural Network (CNN). Aplikasi ini dirancang untuk meningkatkan keefektifan pembelajaran bagi siswa tunarungu di SLB-BC Mahardika Depok, memudahkan pengajar dalam penyampaian materi, dan memfasilitasi siswa untuk langsung mempraktikkan materi yang dipelajari. Dengan teknologi CNN, aplikasi ini menunjukkan kemajuan signifikan dalam pembelajaran bahasa isyarat. Tujuan penelitian ini adalah untuk mempermudah siswa dalam memahami materi yang diberikan melalui aplikasi mobile berbasis bahasa isyarat, dan

berkontribusi pada pengembangan metode pembelajaran yang lebih inovatif.

2. Penelitian berjudul "Semantic Segmentation Based on Depth Background Blur" (Li et al., 2022) mengusulkan penggunaan model blur latar untuk mengatasi masalah latar yang kacau dan variasi pencahayaan. Pendekatan ini telah divalidasi pada arsitektur jaringan saraf yang sudah ada, menunjukkan peningkatan kualitas segmentasi pada dataset seperti PASCAL VOC 2012 dan CamVid. Analisis menunjukkan bahwa model blur latar dapat meningkatkan kualitas segmentasi tanpa perlu merubah arsitektur jaringan yang ada.
3. Penelitian berjudul "Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network" (Das et al., 2020) menemukan bahwa model berbasis Convolutional Neural Network (CNN) untuk mengenali Bahasa Isyarat Amerika (ASL) mencapai tingkat akurasi yang sangat tinggi. Ini mengungguli metode yang ada sebelumnya dan memperlihatkan bahwa CNN sangat efektif untuk klasifikasi gambar dan ekstraksi fitur yang relevan secara otomatis.

1.8 Sistematika Penulisan

Sistematika penulisan skripsi yang digunakan dalam penelitian ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bagian ini menguraikan latar belakang pemilihan topik, definisi dan pembatasan masalah, objektif, kegunaan, dan metode penelitian, studi terkait, serta kerangka penulisan skripsi.

BAB 2 LANDASAN TEORI

Bab ini merupakan beberapa teori yang relevan dengan penelitian dibahas termasuk pengantar Bahasa Isyarat, Sistem Isyarat Bahasa Indonesia, Kecerdasan Buatan, Deep Learning, Machine Learning, Convolutional Neural Network (CNN), Computer Vision, dan Gaussian Blur.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini membahas analisis terhadap algoritma yang digunakan dan perancangan diagram yang diperlukan seperti diagram alir (*flowchart*).

BAB 4 IMPLEMENTASI DAN PENGUJIAN

bab ini menjelaskan implementasi dan pengujian sistem sesuai dengan desain yang telah disiapkan sebelumnya.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari pembahasan di setiap bab dan saran dari peneliti untuk penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Bahasa Isyarat

Bahasa isyarat adalah bentuk bahasa kompleks yang mirip dengan bahasa lisan dalam struktur kalimatnya. Individu dengan gangguan pendengaran (tunarungu) dan gangguan bicara (tunawicara) menggunakan bahasa isyarat untuk berkomunikasi. Bahasa isyarat memungkinkan komunikasi visual tanpa menggunakan sistem auditori. Melalui gerakan tangan, ekspresi wajah, dan bahasa tubuh tunarungu dan tunawicara dapat mengekspresikan pikiran dan emosi mereka. Bahasa isyarat memiliki elemen seperti fonologi, morfologi, sintaksis, dan semantik dalam bahasa yang membuat komunikasi efektif antara tunarungu dan tunawicara dengan orang lain (Nur et al., 2022).

Bahasa isyarat memungkinkan tunarungu dan tunawicara untuk berkomunikasi dengan orang lain. Perkembangan teknologi membuat penelitian tentang pengenalan bahasa isyarat secara otomatis melalui sistem komputer menjadi penting untuk mempermudah interaksi antara tunarungu dan tunawicara dengan orang lain (Naufal & Kusuma, 2023).

2.1.1 *Tunarungu*

Tunarungu adalah orang yang mengalami kehilangan pendengaran sehingga mengganggu kemampuannya dalam memproses informasi verbal melalui pendengaran. Mereka memiliki kecerdasan normal dan rata-rata, namun kemampuannya seringkali lebih rendah dibandingkan non tunarungu (Iskandar & Supena, 2021).

2.1.2 *Tunawicara*

Tunawicara adalah orang yang mengalami gangguan kesulitan dalam komunikasi verbal. Hal ini disebabkan oleh adanya disfungsi pada organ pendengaran dan organ bicara seperti rongga mulut, lidah, dan pita suara sehingga mengakibatkan perkembangan bahasa terhambat (Akhmad et al., 2021).

2.2 Sistem Isyarat Bahasa Indonesia

SIBI singkatan dari Sistem Isyarat Bahasa Indonesia merupakan sistem bahasa isyarat resmi di Indonesia untuk membantu komunikasi antara tunarungu dan tunawicara dengan non difabel (Naufal & Kusuma, 2023). SIBI merupakan jenis bahasa isyarat yang dipakai di Indonesia dan dijadikan sebagai standar pembelajaran di sekolah luar biasa (SLB) (Fatmawati et al., 2022).

SIBI menggunakan sistem komunikasi visual-gestural yang mencakup pemahaman terhadap struktur dari bahasa isyarat, perbendaharaan kata, aturan gramatikal, serta peningkatan kemampuan komunikasi dalam penerapan bahasa isyarat. SIBI di dasarkan pada gerakan tangan, jari, dan ekspresi wajah sebagai sarana komunikasi. Gerakan-gerakan ini merepresentasikan kata, frasa, atau konsep dalam bahasa Indonesia (Anggraini & Zakaria, 2023).

2.3 Artificial Intelligence

Artificial Intelligence merupakan sebuah paradigma yang digabungkan dengan teknik *Machine Learning* akan berdampak signifikan pada berbagai bidang seperti sains, teknologi, industri, dan kehidupan sehari-hari. *Artificial Intelligence* mengacu pada simulasi kecerdasan manusia oleh suatu sistem atau mesin (Xu et al., 2021).

Tujuan utama dari *Artificial Intelligence* adalah meniru fungsi kognitif manusia untuk memecahkan masalah dan menciptakan sistem yang dapat belajar serta berpikir seperti manusia sehingga perangkat lunak dapat menganalisis data historis, mengekstrak, dan membuat prediksi berdasarkan masukan input data (Holzinger et al., 2019).

2.4 Deep Learning

Deep Learning merupakan sebuah bentuk dari jaringan saraf buatan untuk belajar dari data dan membuat keputusan. Istilah "Deep" dalam Deep Learning mengacu pada beberapa lapisan tempat data diproses oleh jaringan saraf. Jaringan saraf ini dapat secara otomatis mempelajari representasi data pada berbagai tingkat abstraksi, memungkinkan mereka mengekstrak pola dan fitur kompleks dari input. *Deep Learning* mencoba meniru fungsi otak manusia dalam mengolah data di seluruh lapisan node atau neuron yang saling berhubungan untuk mengidentifikasi pola dan membuat prediksi (Alzubaidi et al., 2021).

Deep learning terdiri dari Hidden Layer yang mengolah berbagai metadata berdasarkan hasil output dari neuron yang terletak sebelumnya. Jaringan saraf pada deep learning dibangun dengan struktur hirarki sederhana yang mencakup Multi Layer (Sholawati et al., 2022).

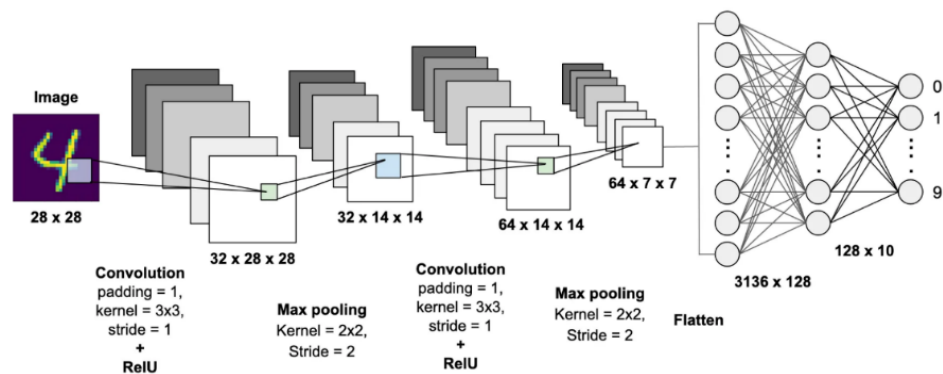
2.5 Machine Learning

Machine learning merupakan sebuah cabang dari *Artificial Intelligence* yang memberikan kemampuan kepada komputer untuk belajar dari pola serta memprediksi data masukan. *Machine learning* dapat mengekstrak dan mengidentifikasi pola-pola yang kompleks serta membantu untuk mengambil keputusan. Dengan menggunakan algoritma dan memanfaatkan jumlah data yang efisien, hal ini menjadi lebih efektif dalam mengambil keputusan yang tepat dalam bidang menganalisis gambar (Badillo et al., 2020).

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah algoritma *Deep Learning* yang dirancang khusus untuk memproses data terstruktur seperti grid seperti gambar dan data deret waktu. *Convolutional Neural Network* (CNN) telah berhasil digunakan di berbagai aspek seperti pengenalan gambar, pengenalan objek, dan analisis gambar (Vaz & Balaji, 2021). Algoritma *Convolutional Neural Network* (CNN) adalah algoritma yang digunakan untuk mempelajari dan mendeteksi fitur pada gambar digital (Sholawati et al., 2022).

Convolutional Neural Network (CNN) memiliki beberapa komponen utama yaitu lapisan konvolusional, lapisan penggabungan, dan lapisan yang terhubung sepenuhnya. Lapisan konvolusional menerapkan filter pada input data yang berguna untuk mengekstrak fitur seperti tepi, tekstur, dan bentuk. Lalu lapisan pengumpulan memiliki fungsi untuk membantu mengurangi dimensi spasial pada fitur yang telah dibuat oleh lapisan konvolusional dan membuat jaringan yang lebih kuat terhadap variasi input data. Kemudian lapisan yang terhubung berfungsi untuk menggabungkan fitur-fitur yang dipelajari dari lapisan sebelumnya untuk membuat prediksi atau klarifikasi (Aljabar & Suharjito, 2020).



Gambar 2.1 *Arsitektur Convolutional Neural Network*

Sumber: Patel, 2019

Pada Gambar 2.1 *Arsitektur Convolutional Neural Network* (CNN) terdiri atas 2 bagian input, ekstraksi fitur, dan klasifikasi. Di dalam lapisan ekstraksi fitur, terdapat beberapa lapisan lain antara lain:

a. *Convolutional layer*

Convolutional Layer merupakan teknik untuk mengekstrak dari gambar menggunakan sebuah filter atau kernel. Teknik konvolusi ini beroperasi dengan cara menggeser filter tersebut melintasi keseluruhan gambar.

Jarak penggeseran filter ini dikenal sebagai satu langkah, dan selama proses ini, fitur-fitur di setiap piksel gambar diekstraksi dengan menambahkan piksel bernilai 0 di sisi matriks. Setiap pergeseran filter melibatkan operasi perkalian matriks antara matriks input dan matriks filter. Proses ini menghasilkan sebuah matriks dua dimensi (Cao et al., 2022).

b. *MaxPooling Layer*

MaxPooling Layer berfungsi Untuk mereduksi ukuran matriks konvolusi.. Tujuan dari proses pooling ini adalah meningkatkan kecepatan proses perhitungan komputasi (Christlein et al., 2019).

Di dalam layer classification terdapat beberapa lapisan yang menjadi komponen di dalamnya, diantaranya:

a. *Flatten*

Flatten merupakan proses konversi matriks yang dihasilkan dari *Feature Learning* dari bentuk dua dimensi menjadi satu dimensi (Jeczmionek & Kowalski, 2021).

b. *Dropout*

Dropout merupakan proses menghapus *neuron* yang dianggap sebagai gangguan secara acak yang bermanfaat untuk menghindari terjadinya *overfitting* (Cai et al., 2019).

c. *Fully Connected Layer*

Fully Connected Layer merupakan proses menggabungkan semua *neuron* ke dalam satu dimensi data yang telah mengalami proses pembelajaran fitur dan telah diratakan menjadi bentuk vektor (Basha et al., 2020).

d. *Fungsi Softmax*

Fungsi Softmax merupakan proses mengkalkulasi nilai probabilitas untuk setiap kelas tujuan bersama semua kelas target. Skala nilai probabilitas yang diperoleh dari penggunaan softmax berkisar antara 0 sampai 1 (Zhu et al., 2020).

2.7 Computer Vision

Computer Vision adalah bagian dari *Artificial Intelligence* yang berpusat pada pemberdayaan komputer untuk menguraikan dan mewujudkan dunia visual. Hal ini melibatkan pengembangan algoritma dan teknik yang memungkinkan Machine Learning untuk mengekstrak informasi dari gambar atau video digital. Proses ini sama dengan cara manusia memandang dan menafsirkan data visual (Fan et al., 2020).

2.8 Algoritma Gaussian Blur

Algoritma *Gaussian Blur* adalah metode *blurring* yang menggunakan filter *Gaussian* untuk mengurangi *detail* dan *noise* dalam sebuah gambar. Algoritma *Gaussian Blur* digunakan untuk menciptakan efek blur yang alami pada gambar dengan memperhitungkan ketajaman dari objek dalam gambar. Tujuan dari *Gaussian Blur* adalah untuk memodelkan efek blur pada suatu gambar dengan mengkonvolusi gambar yang tajam dengan *Point Spread Function* (PSF) yang

diwakili oleh kernel *Gaussian* (Hassan et al., 2019). Filter *Gaussian* digunakan untuk mengaburkan bagian-bagian gambar sesuai dengan ketajaman objek. Objek yang ada dalam gambar muncul secara tajam sementara elemen di luar objek tersebut menampilkan efek blur (Shen et al., 2023).

Gaussian Blur diterapkan dengan menggabungkan gambar dengan fungsi *Gaussian*. *Gaussian Blur* berfungsi sebagai teknik dasar dalam pemrosesan gambar untuk meningkatkan kualitas gambar dengan mengurangi *noise* dan *detail* yang tidak diinginkan, proses ini membantu mengaburkan gambar sekaligus mempertahankan konten frekuensi spasial. Hal ini secara efektif mengurangi *noise* pada gambar sehingga menghasilkan gambar yang lebih bersih dan halus (Devi et al., 2023). Berikut ini adalah rumus fungsi gaussian

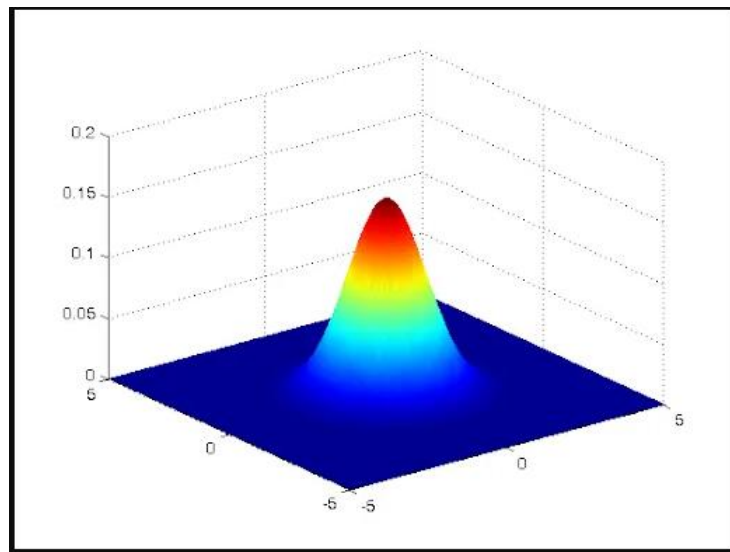
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Dimana:

- $G(x,y)$ adalah Fungsi *Gaussian* memberikan bobot pada setiap piksel di citra berdasarkan jaraknya dari pusat filter.
- σ sigma adalah standar deviasi dari distribusi *Gaussian* ini menentukan seberapa lebar atau sempit bentuk lonceng dari fungsi Gaussian. Semakin besar nilai σ , semakin besar area blur dan semakin halus hasil blurnya.
- e adalah Fungsi eksponensial menurun dengan cepat seiring dengan meningkatnya jarak x dan y dari pusat filter sehingga menghasilkan bobot yang lebih kecil untuk piksel yang lebih jauh dari pusat.
- $\frac{1}{2\pi\sigma^2}$ adalah Koefisien ini adalah faktor normalisasi yang memastikan bahwa total bobot dari semua piksel dalam filter *Gaussian* berjumlah 1. Ini penting untuk mempertahankan kecerahan total citra setelah penerapan blur.

Nilai dari fungsi ini akan membuat matriks konvolusi atau kernel yang akan kita terapkan pada setiap piksel dalam gambar asli. Kernel biasanya cukup kecil semakin besar kernel, semakin banyak komputasi yang harus dilakukan pada setiap piksel. x dan y menentukan delta dari piksel tengah (0, 0). Misalnya, jika radius yang dipilih untuk kernel adalah 3, x dan y akan berkisar dari -3 hingga 3

(inklusif). Standar deviasi memengaruhi seberapa signifikan piksel tetangga dari piksel tengah memengaruhi hasil komputasi (Yang et al., 2020).



Gambar 2.2 *Grafik Fungsi Gaussian*

Sumber : Juniar et al., 2023

Grafik tiga dimensi dari fungsi Gaussian juga dikenal sebagai fungsi distribusi normal. Fungsi Gaussian memiliki bentuk lonceng simetris dengan puncak di pusatnya yang sering disebut sebagai "bell curve". Dalam grafik ini, sumbu x dan y merepresentasikan dua dimensi ruang sementara sumbu z merepresentasikan nilai fungsi Gaussian (Juniar et al., 2023).

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis

Analisis dan perancangan sistem adalah suatu pendekatan yang sistematis untuk merancang dan mengembangkan sistem. Analisis mendefinisikan kebutuhan terkait sistem yang akan dikembangkan. Dengan pendekatan yang sistematis, diharapkan sistem yang dikembangkan dapat memenuhi kebutuhan pengguna secara efektif dan efisien.

3.1.1 *Analisis Masalah*

Penerapan Algoritma Gaussian Blur dalam sistem penerjemah bahasa isyarat dimaksudkan untuk memperbaiki akurasi deteksi gerakan tangan dengan memperhalus gambar dan mengeliminasi gangguan visual. Algoritma ini harus diatur secara cermat untuk tidak menghilangkan detail esensial yang dapat mempengaruhi pemahaman bahasa isyarat. Gaussian Blur dalam sistem penerjemah harus dilakukan pengujian menyeluruh di berbagai kondisi cahaya untuk memastikan bahwa peningkatan tersebut tidak mengurangi kecepatan atau efektivitas sistem. Evaluasi keseluruhan harus mempertimbangkan dampaknya terhadap pengguna, mengoptimalkan antarmuka pengguna, dan memperbaiki keakuratan terjemahan untuk mendukung komunikasi yang lebih efektif.

3.1.2 *Analisis Kebutuhan*

Analisis kebutuhan adalah proses identifikasi, pemahaman, dan spesifikasi kebutuhan yang harus dilengkapi oleh sistem. Analisis kebutuhan dibagi dalam dua bagian utama, yaitu fungsional dan non-fungsional.

1. Kebutuhan Fungsional

Kebutuhan fungsional adalah spesifikasi fitur dan fungsi yang harus dimiliki oleh sistem untuk mencapai tujuan utama. Penelitian ini memiliki kebutuhan fungsional utama, yaitu:

- a. Peningkatan Kualitas Gambar: Gaussian Blur harus mampu mengurangi noise visual dan meningkatkan kejelasan gambar tangan dan gerakan jari, yang kritikal dalam bahasa isyarat.
- b. Akurasi Penerjemahan: Sistem harus mampu mengenali dan menerjemahkan gerakan tangan dan posisi jari dengan akurat menggunakan teknologi pengolahan gambar yang disempurnakan oleh Gaussian Blur.
- c. Adaptasi Terhadap Kondisi Pencahayaan: Gaussian Blur harus dapat menyesuaikan gambar di bawah berbagai kondisi pencahayaan untuk mempertahankan akurasi pengenalan gerakan.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah cara di mana sistem berjalan dan beroperasi dengan seharusnya. Penelitian ini memiliki beberapa kebutuhan non-fungsional, yaitu:

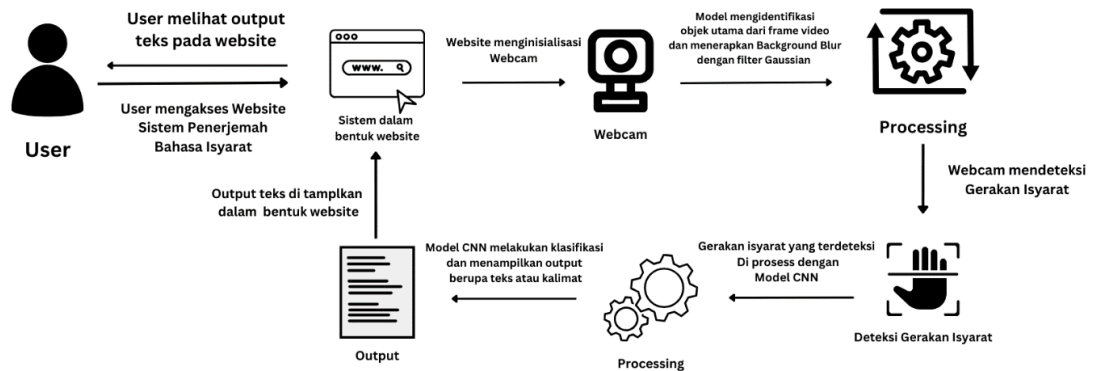
- a. Kecepatan dan Performa: Sistem harus cepat dan responsif dapat beroperasi secara real-time untuk mendukung komunikasi yang lancar.
- b. Kemudahan Penggunaan: Antarmuka pengguna harus intuitif dan mudah digunakan agar memudahkan navigasi dan pemahaman hasil terjemahan.
- c. Kemudahan Integrasi: Algoritma Gaussian Blur harus dapat diintegrasikan dengan mudah ke dalam komponen sistem yang ada.
- d. Penggunaan di Berbagai Kondisi Pencahayaan: Sistem harus efektif di berbagai kondisi pencahayaan, memastikan konsistensi dan keandalan pengenalan gerakan.

3.2 Perancangan Sistem

Sistem yang akan dirancang, dibutuhkan beberapa diagram dengan menggambarkan spesifikasi detail tentang sistem yang direncanakan menggunakan diagram. Selain itu, dipertimbangkan alur proses dari perancangan program untuk memastikan keselarasan dan konsistensi.

3.2.1 Diagram Umum Sistem

Gambaran umum sistem pada penelitian ini digunakan Untuk mengilustrasikan secara menyeluruh sistem yang akan dikembangkan. Gambaran umum sistem akan dijelaskan secara rinci dalam **Gambar 3.1** di bawah ini.

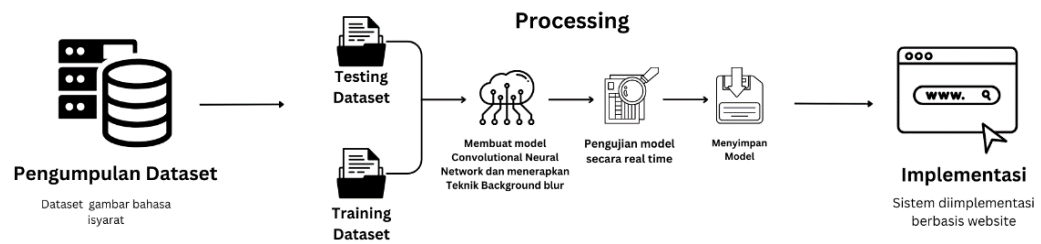


Gambar 3.1 Gambaran Umum Sistem

Pada Gambar diatas, sistem diimplementasikan dalam bentuk website dengan proses:

- 1) *User* pergi ke website sistem penerjemah bahasa isyarat.
- 2) Website memberikan opsi untuk mulai mendeteksi gerakan bahasa isyarat.
- 3) Website meminta akses untuk menggunakan device webcam yang tersedia untuk mendeteksi gerakan.
- 4) Model melakukan pemrosesan terhadap *frame* video dari webcam dan mendeteksi objek utama dalam *frame* serta menerapkan *background blur* dengan filter *Gaussian Blur*.
- 5) Webcam yang sudah menerapkan *background blur* mendeteksi gerakan isyarat.
- 6) Model *Convolutional Neural Network* (CNN) melakukan pemrosesan data terhadap gerakan yang di deteksi dengan webcam.
- 7) Gerakan bahasa isyarat yang terdeteksi akan di proses dan di terjemahkan ke dalam website dalam bentuk kata.

Dalam Gambaran Umum Sistem, terdapat Rancangan Sistem yang meliputi setiap tahapan pembuatan sistem seperti pada **Gambar 3.2**



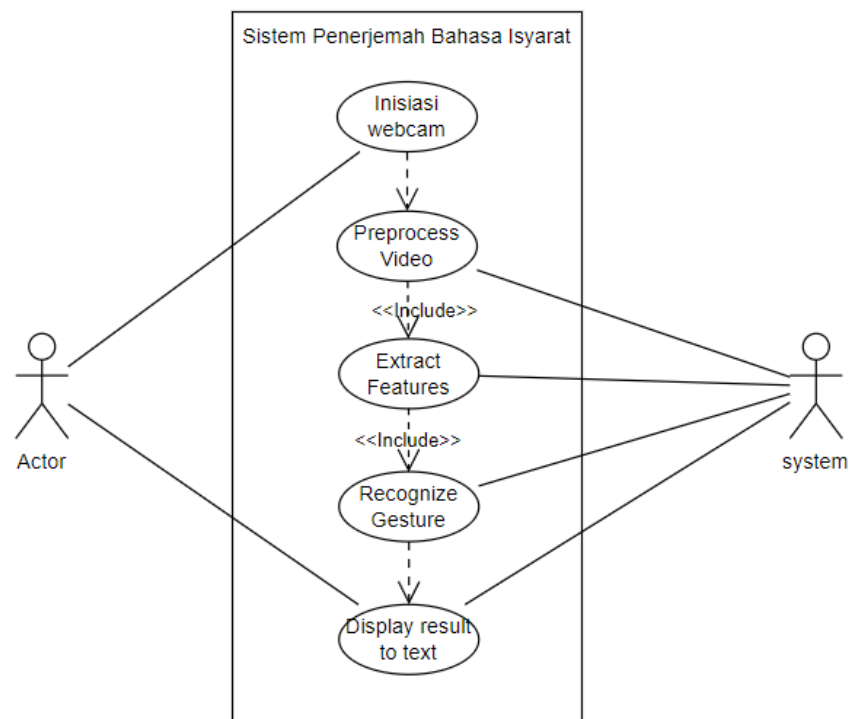
Gambar 3.2 Rancangan Sistem

Proses rancangan sistem pada Gambar 3.2 sebagai berikut:

1. Dataset yang dikumpulkan berupa dataset gambar dari gerakan isyarat yang dikumpulkan oleh peneliti.
2. Pada tahap processing dilakukan tahap pembagian dataset menjadi Training dataset dan Testing dataset, kemudian dilakukan proses pembuatan model dengan *Convolutional Neural Network* (CNN) untuk pendeteksian gerakan isyarat dan *Gaussian Blur* untuk menerapkan teknik pengaburan latar belakang. Setelah dibangun model, akan dilakukan pengujian data secara *real-time* untuk mendapatkan hasil akurasi prediksi yang baik. Model yang sudah teruji dengan baik akan dilakukan penyimpanan untuk bisa diimplementasikan pada sebuah website.
3. Pada Implementasi, dilakukan implementasi sistem dalam bentuk website. Website ini akan digunakan oleh tunarungu dan tunawicara. Hasil akhir yang terdapat pada website merupakan hasil dari penerjemahan gerakan isyarat berupa kata.

3.2.2 Use Case Diagram

Use Case Diagram pada sistem ini digunakan untuk menggambarkan interaksi antara sistem dan aktor-aktor eksternal yang terlibat dalam penggunaan sistem. Use Case Diagram sistem akan dijelaskan secara rinci dalam **Gambar 3.3** di bawah ini.

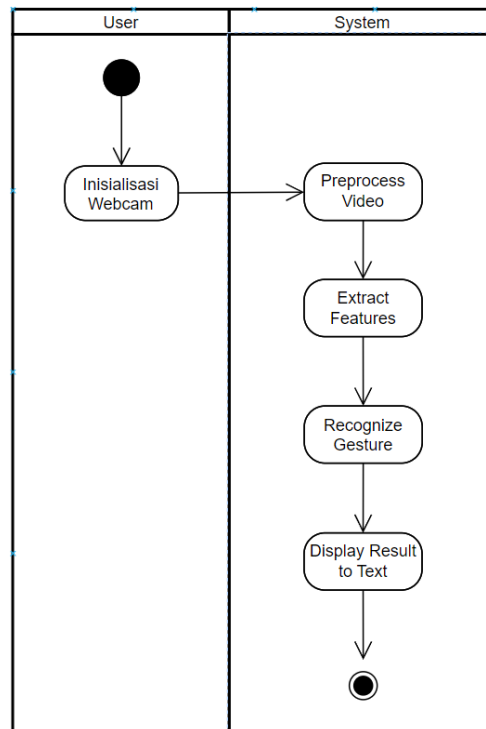


Gambar 3.3 *Use Case Diagram*

Pada Gambar 3.3 merupakan *Use Case diagram* yang menjelaskan sistem penerjemah bahasa isyarat yang terdiri dari beberapa tahap utama yang melibatkan interaksi antara sistem dan aktor. Pertama, sistem menginisiasi webcam untuk merekam video dari aktor. Selanjutnya, video tersebut diproses yang meliputi ekstraksi fitur dari video yang telah diproses. Setelah fitur diekstraksi, sistem mengenali gerakan tangan atau gesture dari bahasa isyarat yang dilakukan oleh aktor. Hasil pengenalan gerakan ini kemudian ditampilkan dalam bentuk teks.

3.2.3 *Activity Diagram*

Activity Diagram pada sistem ini digunakan untuk menggambarkan aliran kerja atau aktivitas dalam sistem. Activity Diagram sistem akan dijelaskan secara rinci dalam **Gambar 3.4** di bawah ini.



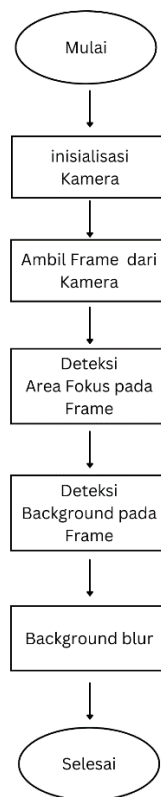
Gambar 3.4 *Activity Diagram*

Pada Gambar 3.4 merupakan *Activity Diagram* yang menjelaskan proses kerja sebuah sistem penerjemah bahasa isyarat menjadi teks. Dimulai dengan pengguna yang mengaktifkan webcam. Kemudian, sistem memproses video yang diterima dari webcam tersebut. Langkah selanjutnya adalah ekstraksi fitur, sistem mengidentifikasi aspek-aspek penting dari gerakan tangan pengguna. Setelah fitur-fitur ini diekstraksi, sistem mengenali gestur yang ditunjukkan. Hasil dari pengenalan gestur ini ditampilkan dalam bentuk teks kepada pengguna.

3.3 *Flowchart* (Diagram Alir)

Flowchart adalah sebuah diagram yang digunakan untuk merepresentasikan urutan langkah-langkah dan keputusan yang diperlukan untuk melakukan suatu proses. *Flowchart* dapat digunakan untuk mendokumentasikan, mempelajari, merencanakan, meningkatkan, dan berkomunikasi mengenai proses yang kompleks dalam bentuk diagram yang mudah dipahami.

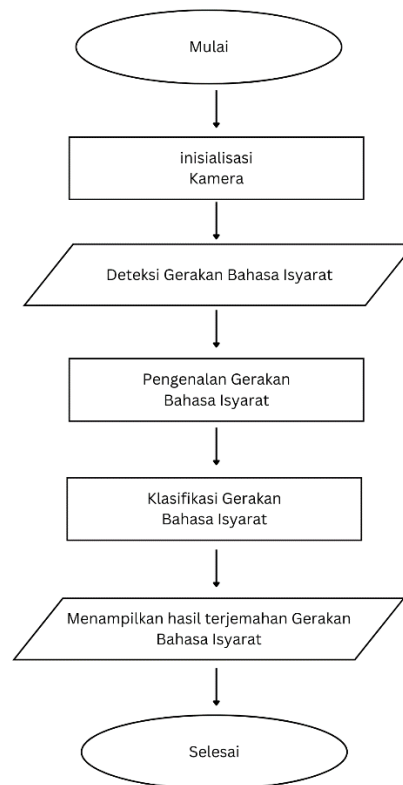
3.3.1 Flowchart Background blur



Gambar 3.5 Flowchart Background blur

Pada *flowchart* Gambar 3.5 menunjukkan rangkaian proses untuk menerapkan *background blur* atau pengaburan latar belakang secara *real time*. Proses dimulai dengan proses menginisialisasi kamera untuk mendapatkan input video secara *real time*. Kemudian dilakukan proses mengambil *frame* dari input kamera. Lalu, dilakukan deteksi area fokus pada *frame*. Setelah itu, dilakukan deteksi *background* yang perlu di blur pada *frame*. Langkah berikutnya menerapkan proses *background blur* dengan menerapkan filter *Gaussian Blur*.

3.3.2 Flowchart Sistem Penerjemah Bahasa Isyarat



Gambar 3.6 Flowchart Sistem Penerjemah Bahasa Isyarat

Proses yang dilakukan pada Gambar 3.6 merupakan *flowchart* Sistem Penerjemah Bahasa isyarat yang akan dibuat. Proses diawali dengan inisialisasi kamera. Selanjutnya, kamera akan mendeteksi gerakan bahasa isyarat. Setelah itu, model sistem akan mengenali gerakan bahasa isyarat dan mengklasifikasikan gerakan bahasa isyarat tersebut sesuai dengan dataset yang sudah dikumpulkan. Lalu, sistem akan menampilkan hasil terjemahan gerakan bahasa isyarat dalam bentuk teks.

3.4 Analisis Dataset

Dataset yang digunakan berupa dataset gambar yang dikumpulkan menggunakan webcam dan diambil dengan bantuan website Teachable Machine Learning. Website ini merupakan website untuk pengambilan dataset gambar yang memotret gerakan bahasa isyarat secara otomatis. Dataset terdiri dari 3.480 gambar gerakan bahasa isyarat, 2.420 gambar untuk data latih dan 1.040 untuk data validasi dengan 20 kategori/kelas.

3.5 Perancangan Pre-processing Dataset

Pre-Processing merupakan teknik untuk mengolah data yang tidak terorganisir menjadi data yang terorganisir dan siap diolah. Tahapannya yaitu Data Resizing, Image Annotation dan Augmentasi Data.

3.5.1 *Data Resizing*

Data resizing merupakan sebuah proses citra diubah ukurannya agar sesuai dengan dimensi yang diinginkan. Proses ini bertujuan untuk menstandarisasi dimensi semua citra yang masuk agar sesuai sebelum digunakan untuk pelatihan model karena pelatihan model memerlukan input yang memiliki dimensi yang seragam.

3.5.2 *Image Annotation*

Anotasi gambar melibatkan penambahan titik-titik pada gambar yang digunakan untuk membantu pelatihan model machine learning dalam. Proses ini untuk menandai dan mengenali bagian tertentu dari gambar. Berikut ini merupakan tahapan dari proses *Image Annotation* :

1. Deteksi *Landmark*

Menggunakan library MediaPipe untuk melakukan deteksi landmark pada tangan, wajah, dan pose tubuh. Citra diubah dari format BGR ke RGB proses ini diperlukan library MediaPipe untuk proses deteksi.

2. Inisialisasi Citra Anotasi

Setelah deteksi landmark berhasil lalu menginisialisasi citra baru yang awalnya kosong (diisi dengan warna putih) untuk menandai anotasi secara lebih jelas.

3. Penggambaran *Landmark*

Landmark yang ditemukan kemudian dirender pada citra yang telah disiapkan. Proses ini melibatkan mengambil citra, landmark yang terdeteksi, serta spesifikasi visual termasuk warna dan ketebalan garis untuk mengilustrasikan hubungan antara landmark. Spesifikasi visual yang variatif untuk tangan kanan, tangan kiri, wajah, dan pose tubuh memungkinkan diferensiasi tipe landmark yang diannotasi pada citra.

4. Penyimpanan Citra yang Telah Diannotasi

Setelah anotasi landmark selesai citra kemudian disimpan ke dalam direktori output yang telah ditentukan dalam kode. Citra tersebut menunjukkan semua landmark penting yang telah diberi tanda yang dapat digunakan sebagai bagian dari dataset pelatihan untuk model machine learning.

3.5.3 *Data Augmentasi*

Augmentasi diterapkan pada dataset pelatihan untuk menambahkan variasi dan menghindari model dari belajar hanya pada representasi data yang tertentu. Hal ini bertujuan untuk memperbaiki kemampuan model dalam menghadapi data baru yang belum dikenal sebelumnya (generalisasi).

Augmentasi yang dilakukan secara dinamis jadi tidak ada penambahan dari dataset latih tetapi selama pelatihan berlangsung model akan melihat variasi gambar yang lebih banyak. Augmentasi tidak diaplikasikan pada dataset validasi karena tujuan dari dataset ini adalah untuk menguji efektivitas model pada data yang tidak dimodifikasi. Augmentasi membantu model menjadi lebih robust terhadap variasi dalam data input yang sebenarnya. Berikut adalah detail dari augmentasi yang diterapkan:

1. *Rescaling*

Proses mengubah skala intensitas piksel citra menjadi rentang 0 hingga 1 merupakan praktik umum untuk memudahkan model dalam memproses citra.

2. *Zoom Range*

Pada proses ini citra diperbesar atau diperkecil secara acak dalam kisaran yang ditentukan antara 0,5 dan 1,5.

3. *Width Shift dan Height Shift*

Pada proses ini citra digeser secara horizontal atau vertikal hingga batas tertentu dengan 20% lebar dan 30% tinggi citra.

3.6 Perancangan Teknik *Background Blur*

3.6.1 *Tensorflow.js*

TensorFlow.js merupakan pustaka yang dirancang untuk mengembangkan serta mengeksekusi algoritma pembelajaran mesin menggunakan JavaScript yang memfasilitasi pembuatan dan penerapan model Machine Learning di browser web. Library ini adalah bagian dari TensorFlow yang lebih luas yang menyediakan API yang memungkinkan kompatibilitas lintas-platform antara Python dan JavaScript (Smilkov et al., 2019).

Library *Tensorflow.js* digunakan untuk menerapkan *background blur* secara real-time karena di dalam *Tensorflow.js* terdapat model yang dapat mensegmentasi bagian person dengan latar belakang sehingga dapat memisahkan antara latar belakang dan person.

3.6.2 *BodyPix*

BodyPix merupakan model segmentasi yang beroperasi secara real-time dan dapat mengisolasi serta mengenali berbagai bagian tubuh manusia dalam sebuah gambar atau video. Model ini memungkinkan pemisahan tubuh manusia ke dalam komponen seperti kepala, tangan, dan kaki tanpa kebutuhan akan green screen atau latar belakang khusus (Bailey et al., 2019).

1. Arsitektur *MobileNetV1*

Arsitektur yang digunakan model BodyPix untuk segmentasi manusia menggunakan arsitektur MobileNetV1. Arsitektur ini menggunakan teknik *separable depthwise convolution* yang mengurangi jumlah operasi dan parameter sehingga membuatnya menjadi ringan dan cepat tanpa mengorbankan terlalu banyak akurasi.

2. *Model Size*

Model Size menggunakan model MobileNetMultiplier yang merupakan pengaturan konfigurasi untuk model Arsitektur *MobileNet*.

Model size memiliki beberapa nilai ukuran model diantaranya 0.25 atau 0.50 untuk kecepatan yang lebih tinggi tetapi mengurangi akurasi dan 0.75 atau 1.00 untuk akurasi segmentasi yang lebih tinggi namun mengurangi kecepatan pemrosesan.

Pada perancangan ini ukuran model di terapkan menjadi 0.75 agar sistem bekerja secara optimal dengan sedikit pengorbanan akurasi dari segmentasi.

3. *OutputStride*

OutputStride merupakan downsampling dari citra sehingga jika *OutputStridenya* lebih tinggi seperti 16 atau 32 maka akan menghasilkan output yang lebih cepat namun kurang akurat karena setiap stride akan melompati banyak piksel input-an. *OutputStride* yang lebih rendah seperti 8 atau bahkan 4 akan menghasilkan output yang lebih akurat tetapi tidak optimal pada sistem.

Pada perancangan ini *OutputStride* akan di terapkan menjadi 16.

4. *QuantBytes*

QuantBytes merupakan penentuan jumlah byte yang akan digunakan untuk menyimpan bobot model yang telah dikuantisasi. Kuantisasi adalah teknik yang mengurangi presisi numerik bobot dan aktivasi model untuk mengurangi ukuran model dan mempercepat inferensi tanpa mengurangi akurasi secara signifikan. *QuantBytes* memiliki beberapa nilai yang umum digunakan diantaranya:

- 1 byte menggunakan 8-bit per bobot menghasilkan ukuran model yang sangat kecil dan waktu inferensi yang cepat, tetapi bisa berpotensi kehilangan informasi yang lebih besar, yang mungkin mempengaruhi akurasi model.
- 2 byte menggunakan 16-bit per bobot memberikan presisi yang cukup untuk banyak aplikasi tanpa terlalu banyak overhead.
- 4 byte menggunakan 32-bit per bobot dengan presisi floating point standar memberikan akurasi yang hampir sama dengan model non-kuantisasi dan memberikan

beberapa peningkatan dalam efisiensi biasa digunakan untuk aplikasi yang memerlukan akurasi tinggi tanpa keterbatasan komputasi.

Pada perancangan ini *QuantBytes* akan di terapkan menjadi 2 byte karena memerlukan kecepatan dengan mengorbankan sedikit akurasi.

3.7 Perancangan Interface Website

Pada perancangan interface website menggunakan beberapa framework yaitu Web Framework Flask untuk mengintegrasikan bahasa pemrogramana python kedalam bentuk website dan Bootstrap Framework untuk mengintegrasikan CSS kedalam website.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Setelah melakukan analisis dan perancangan, langkah berikutnya adalah menguji perancangan sistem dengan dataset.

4.1.1 Dataset

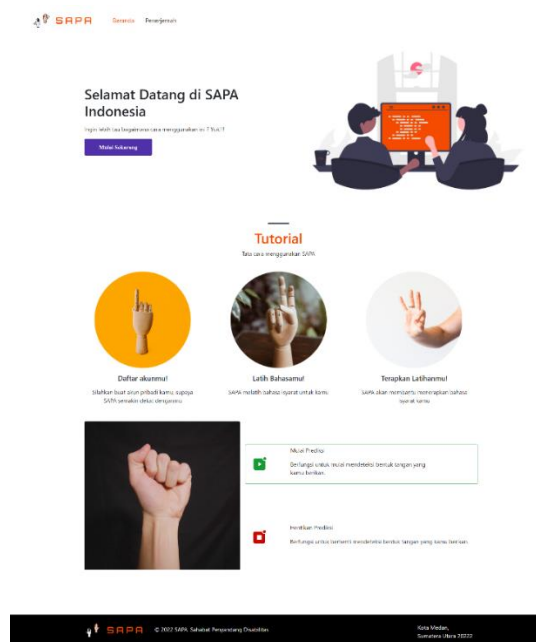
Berikut adalah Dataset yang dikumpulkan antara lain:

1. Dataset berupa Gambar gerakan Bahasa Isyarat yang diambil oleh peneliti sebanyak 20 kata dan setiap kelas terdapat 175 Gambar gerakan bahasa isyarat.

4.2 Implementasi Interface Website

4.2.1 Halaman Home

Home page atau halaman home adalah layar pertama yang muncul pada website, halaman ini terdapat petunjuk dan tutorial penggunaan website. Tampilan dari Home terdapat pada **Gambar 4.1**.

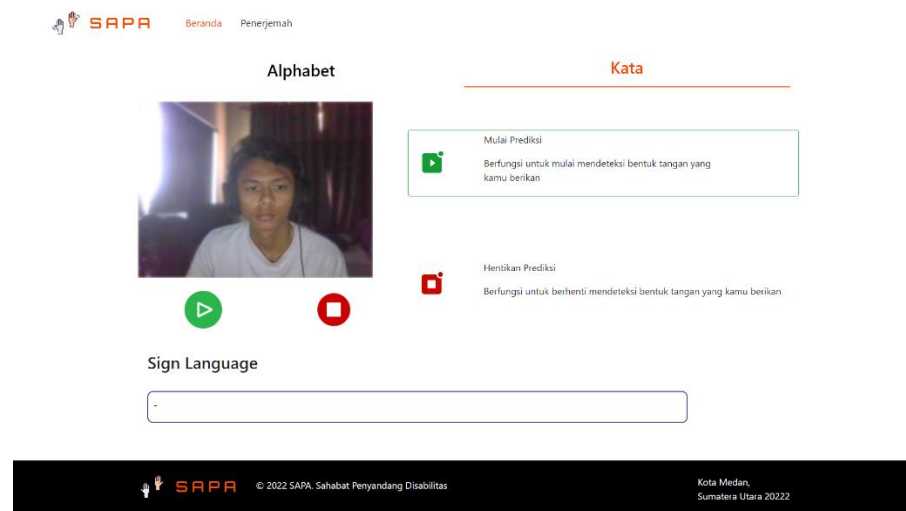


Gambar 4.1 Halaman Home

Pada halaman Home ini Terdapat tombol “Mulai Sekarang” dan pada navigation bar terdapat link “penerjemah” keduanya berfungsi untuk pergi ke halaman penerjemah.

4.2.2 Halaman Penerjemah

Pada halaman Penerjemah merupakan tampilan kedua website dimana di halaman ini berfungsi untuk. Tampilan dari Home terdapat pada Gambar 4.2.



Gambar 4.2 Halaman Penerjemah

Pada halaman Penerjemah ini terdapat icon Play berwarna hijau dan icon Stop berwarna merah. Pada kolom Sign Language merupakan hasil output dari gerakan bahasa isyarat yang akan di deteksi melalui kamera dengan menekan icon play untuk prediksi dan icon stop untuk berhenti prediksi. terdapat link “Alphabet” yang berfungsi untuk pergi ke halaman penerjemah alphabet.

4.3 Implementasi Teknik *Background Blur*

4.3.1 *Tensorflow.js*

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/body-pix"></script>
```

Gambar 4.3 Import Library *Tensorflow.js*

Pada Gambar 4.3 merupakan salah satu cara untuk menyertakan library *Tensorflow* dan model *BodyPix* dari sisi aplikasi web dan server. Library tersebut akan digunakan untuk melakukan segmentasi sehingga dapat memisahkan antara latar belakang dan tubuh.

```
var canvas = document.getElementById('canvas')
var context = canvas.getContext('2d')
const video = document.querySelector('#videoElement')
var dummy = document.querySelector('#dummy')
video.width = 400
video.height = 300

if (navigator.mediaDevices.getUserMedia) {
  navigator.mediaDevices
    .getUserMedia({ video: true })
    .then(function (stream) {
      video.srcObject = stream
      video.play()
    })
    .catch(function (error) {})
}
```

Gambar 4.4 *setupCamera*

Tahap selanjutnya pada bagian `<script>` membuat sebuah kode untuk mengakses webcam pengguna dan menampilkannya pada elemen `<video>` pada halaman web. Pada Gambar 4.4 Kode ini mengatur elemen `'video'` dan `'canvas'` di halaman web serta menangkap video dari kamera pengguna. Pertama, elemen canvas konteks 2D dari canvas dan elemen `'video'` diambil dari Document Object Model (DOM) dengan ukuran video diatur menjadi 400x300 piksel.

Setelah itu dengan kode `'navigator.mediaDevices.getUserMedia'` aplikasi meminta izin untuk mengakses kamera. Jika izin diberikan maka video dari kamera ditetapkan sebagai sumber video dan diputar dalam elemen `'video'`. Jika ada kesalahan akan ditangani oleh fungsi `'catch'` meskipun tidak ada penanganan kesalahan yang spesifik dalam kode ini. Hasil akhirnya adalah video langsung dari kamera pengguna yang ditampilkan pada halaman web.

4.3.2 Bodypix

```
// Fungsi untuk memulai segmentasi tubuh
async function startSegmentation() {
  const video = await setupCamera();
  video.play();
}
```

Gambar 4.5 *startSegmentation*

Pada gambar 4.5 merupakan fungsi untuk memulai segmentasi tubuh dengan nama fungsi '*startSegmentation()*', setelah itu fungsi akan menginisialisasi dan mendapatkan stream video dari webcam dengan fungsi '*video.play()*'.

```
let net;
async function setupBodyPix() {
  net = await bodyPix.load({
    architecture: 'MobileNetV1', // 'ResNet50' is another option
    outputStride: 16, // 8, 16, or 32
    multiplier: 0.75, // 0.50, 0.75, or 1.0
    quantBytes: 2 // 1, 2, or 4
  });
  console.log('BodyPix model loaded.');
}

setupBodyPix();
```

Gambar 4.6 *updateSegmentation*

Pada gambar 4.6 merupakan tahapan untuk memuat model BodyPix dengan menggunakan *architecture MobileNetV1* dengan *outputStride 16*, *multiplier 0.75* dan *quantBytes 2*.

4.3.3 Segmentasi dan Efek Blur

```
async function blurBackground() {
  const segmentation = await net.segmentPerson(video);
  const backgroundBlurAmount = 2;
  const edgeBlurAmount = 1;
  const flipHorizontal = true;

  bodyPix.drawBokehEffect(
    outputCanvas, video, segmentation, backgroundBlurAmount,
    edgeBlurAmount, flipHorizontal);
}
```

Gambar 4.7 Segmentasi dan Efek Blur

Pada gambar 4.7 merupakan Fungsi *blurBackground* adalah fungsi asinkron yang menggunakan pustaka BodyPix untuk memberikan efek blur pada latar belakang video. Fungsi ini pertama-tama melakukan segmentasi pada video untuk memisahkan orang dari latar belakang dengan metode *'net.segmentPerson(video)'*.

Setelah itu, ditetapkan nilai *'2'* untuk jumlah blur latar belakang *'backgroundBlurAmount'*, nilai *'1'* blur pada tepi *'edgeBlurAmount'*, dan *'true'* pada video dibalik secara horizontal *'flipHorizontal'*. Kemudian, fungsi *'bodyPix.drawBokehEffect'* digunakan untuk menggambar efek bokeh pada elemen canvas *'outputCanvas'* berdasarkan segmentasi dan parameter blur yang telah ditentukan sehingga menghasilkan video dengan latar belakang yang kabur sementara orang di dalam video tetap fokus.

4.4 Pengujian Sistem

4.4.1 Pre-Processing Data

1. Data Resizing

Data Resizing digunakan untuk melakukan perubahan ukuran gambar agar mempermudah proses training gambar serta dapat mengurangi waktu pemrosesan klasifikasi gambar.

```

DESIRED_HEIGHT = 224
DESIRED_WIDTH = 224
def resize_and_show(image):
    h, w = image.shape[:2]
    if h < w:
        img = cv2.resize(image, (DESIRED_WIDTH, math.floor(h/(w/DESIRED_WIDTH))))
    else:
        img = cv2.resize(image, (math.floor(w/(h/DESIRED_HEIGHT)), DESIRED_HEIGHT))
    cv2.imshow(img)

```

Gambar 4.8 *Data Resizing*

Pada proses ini gambar akan diubah dari berbagai ukuran piksel menjadi menjadi ukuran 224x224 piksel. Dengan menggunakan bantuan library OpenCV untuk melakukan resizing gambar, serta modul *math* untuk melakukan perhitungan matematika. Terutama fungsi *'cv2.resize()'* dari OpenCV digunakan untuk mengatur dimensi

gambar agar sesuai dengan preferensi tertentu (*DESIRED_HEIGHT* x *DESIRED_WIDTH*). Sedangkan '*math.floor()*' digunakan untuk pembulatan nilai ke bawah, biasanya untuk mendapatkan ukuran yang tepat saat melakukan perhitungan proporsional untuk mempertahankan aspek rasio gambar asli.

2. Image Annotation

Pada tahap ini merupakan tahap untuk mendeteksi dan mengannotasi landmark pada pose tubuh, tangan, dan wajah dari dataset gambar yang diberikan. Setelah itu dataset gambar yang sudah di anotasi akan disimpan untuk analisis kebutuhan training model.

```
[ ] mp_holistic = mp.solutions.holistic
    mp_hands = mp.solutions.hands
    mp_drawing = mp.solutions.drawing_utils
    mp_drawing_styles = mp.solutions.drawing_styles
```

Gambar 4.9 Modul *Mediapipe*

Dengan menginisialisasi modul Holistic '*mp_holistic = mp.solutions.holistic*' dari MediaPipe yang menyediakan fungsi untuk mendeteksi pose tubuh secara keseluruhan diantaranya pose, tangan, dan wajah.

Mengimpor modul Hands dengan '*mp_hands = mp.solutions.hands*' dari *MediaPipe* yang digunakan khusus untuk deteksi dan pelacakan tangan serta mendeteksi titik-titik penting (landmarks) pada tangan termasuk jari-jari dan telapak tangan.

Modul *drawing_utils* seperti '*draw_landmarks*' dapat menggambar titik-titik (landmark) pada gambar dataset. '*drawing_styles*' digunakan bersamaan dengan '*drawing_utils*' untuk memberikan variasi visualisasi pada hasil deteksi landmarks.

```

# Input and output folder paths
input_root_folder = '/content/drive/MyDrive/RisetSLI/DATASETK'
output_root_folder = '/content/drive/MyDrive/RisetSLI/LANDMARK/LDATASETK'

# Iterate through subfolders in the input root folder
for subfolder in os.listdir(input_root_folder):
    # Construct the full path to the subfolder
    input_folder_path = os.path.join(input_root_folder, subfolder)

    # Check if it's a directory
    if os.path.isdir(input_folder_path):
        # Create the corresponding subfolder in the output root folder
        output_folder_path = os.path.join(output_root_folder, subfolder)
        os.makedirs(output_folder_path, exist_ok=True)

        # Iterate through files in the subfolder
        image_files = [f for f in os.listdir(input_folder_path) if os.path.isfile(os.path.join(input_folder_path, f))]

        for image_file in image_files:
            # Construct the full path to the input image
            input_image_path = os.path.join(input_folder_path, image_file)

            # Read the input image
            image = cv2.imread(input_image_path)

            with mp_hands.Hands(min_detection_confidence=0.5, min_tracking_confidence=0.5) as hands:
                results = hands.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
                hand_found = bool(results.multi_hand_landmarks)
                if hand_found:
                    annotated_image = image.copy()
                    annotated_image = np.empty(annotated_image.shape)
                    annotated_image.fill(255)
                    for hand_landmarks in results.multi_hand_landmarks:
                        mp_drawing.draw_landmarks(annotated_image, hand_landmarks, mp_hands.HAND_CONNECTIONS,
                                                    mp_drawing.DrawingSpec(
                                                        color=(0, 0, 255), thickness=4, circle_radius=3),
                                                    mp_drawing.DrawingSpec(
                                                        color=(0, 255, 0), thickness=4, circle_radius=2),
                                                    )
                    resize_and_show(annotated_image)

            # Save the annotated image to the output folder
            output_image_path = os.path.join(output_folder_path, f"annotated_{image_file}")
            cv2.imwrite(output_image_path, annotated_image)
            print(f"Annotated image saved to: {output_image_path}")

```

Gambar 4.10 Proses Anotasi Gambar Dataset

MediaPipe Holistic digunakan untuk mengidentifikasi dan mengannotasi landmark tangan pada gambar Dataset yang ada. Pertama, objek 'Hands' dari MediaPipe diinisialisasi dengan dua parameter: '*min_detection_confidence*' dan '*min_tracking_confidence*' yang masing-masing diatur ke 0.5. Parameter '*min_detection_confidence*' menetapkan ambang batas minimum kepercayaan untuk menganggap deteksi tangan valid, sedangkan '*min_tracking_confidence*' menetapkan ambang batas minimum kepercayaan untuk melacak tangan yang terdeteksi.

Gambar input yang awalnya dalam format BGR (format default OpenCV) di konversi format RGB menggunakan '*cv2.cvtColor(image, cv2.COLOR_BGR2RGB)*' karena MediaPipe memproses gambar dalam format RGB. Gambar yang telah dikonversi kemudian diproses oleh metode '*hands.process()*' yang melakukan deteksi tangan pada gambar tersebut. Hasil deteksi disimpan dalam objek '*results*'.

Jika tangan terdeteksi '*results.multi_hand_landmarks*' tidak '*None*' gambar akan disalin dan anotasi ditambahkan pada salinan

gambar tersebut. Anotasi mencakup menggambar landmark tangan pada salinan gambar menggunakan `'mp_drawing.draw_landmarks()'`. Setelah gambar dianotasi, gambar ini disimpan ke folder output dengan nama file yang telah dimodifikasi untuk menunjukkan bahwa gambar tersebut telah dianotasi. Proses penyimpanan dilakukan menggunakan `'cv2.imwrite()'` yang menyimpan gambar yang telah dianotasi ke path output yang telah ditentukan.

3. Augmentasi Data

Pada tahap ini dataset akan di augmentasi menggunakan *ImageDataGenerator* dari *TensorFlow* yang digunakan untuk mempersiapkan data gambar untuk training model.

```
train_datagen = ImageDataGenerator(
    rescale = 1./255,
    fill_mode='nearest',
    zoom_range=[0.5, 1.5],
    width_shift_range=0.2, height_shift_range=0.3
)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_folder,
    target_size=(224, 224),
    color_mode='grayscale',
    class_mode='categorical',
    shuffle=True,
)

test_generator = test_datagen.flow_from_directory(
    test_folder,
    target_size=(224, 224),
    color_mode='grayscale',
    class_mode='categorical',
    shuffle=True
)
```

Gambar 4.11 Proses Augmentasi

Untuk data training data generator ini mengkonfigurasi penskalaan nilai piksel, pembesaran acak dan pergeseran posisi gambar melalui

`'rescale=1./255, zoom_range=[0.5, 1.5], width_shift_range=0.2'`, dan `'height_shift_range=0.3'`. Sehingga memungkinkan gambar untuk disesuaikan dalam skala, digeser secara horizontal dan vertikal, serta diputar untuk memperkenalkan variasi dan meningkatkan kemampuan generalisasi model.

Untuk testing data augmentasi hanya melibatkan penskalaan nilai piksel untuk memastikan bahwa model dievaluasi secara konsisten pada data yang tidak diubah. Penggunaan *'flow_from_directory'* memudahkan pemuatan dan transformasi gambar secara otomatis dari direktori yang ditentukan, mempersiapkannya dalam batch untuk proses pelatihan dan validasi dengan efisien.

4.4.2 Pengujian Model

Model dibuat sebagai model Sequential dan di train menggunakan TensorFlow dan Keras yang merupakan tumpukan lapisan di mana setiap lapisan memiliki tepat satu input tensor dan satu output tensor.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu', input_shape=(224, 224, 1)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(20, activation='softmax')
])

model.compile(loss='categorical_crossentropy',
              optimizer='RMSprop',
              metrics=['accuracy'])

model.fit(
    train_generator,
    steps_per_epoch=16,
    epochs=50,
    validation_data=test_generator,
    validation_steps=8,
    callbacks = [visualisasi_pelatihan],
    verbose=2)
```

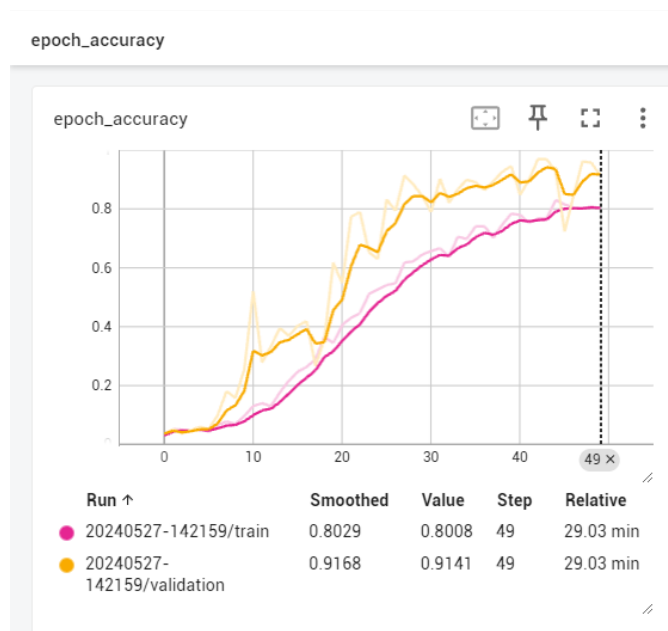
Gambar 4.12 Training Model

Model ini memiliki beberapa lapisan *Conv2D* dengan aktivasi *'relu'*. Lapisan-lapisan ini bertujuan untuk mengekstrak fitur dari Gambar dengan menerapkan filter konvolusional. Setiap lapisan konvolusional diikuti oleh lapisan *MaxPooling2D* berfungsi untuk mengurangi dimensi spasial dari representasi fitur, sehingga mengurangi jumlah parameter dan komputasi dalam jaringan, juga membantu menghindari overfitting.

Lapisan *Dropout* dengan rate 0.4 digunakan untuk mengurangi overfitting lebih lanjut dengan secara acak mengatur output fitur dari lapisan sebelumnya menjadi nol selama pelatihan.

Flatten digunakan untuk mengubah tensor multi-dimensi menjadi vektor satu dimensi. Kemudian, dua lapisan *Dense* digunakan dimana lapisan terakhir menggunakan fungsi aktivasi *softmax* untuk output prediksi kelas dengan asumsi model ini mengklasifikasikan Gambar ke dalam 20 kelas yang berbeda.

Dalam hal ini data akan dilatih dengan beberapa parameter seperti menggunakan *Adam* sebagai *Optimizer* dan *Categorical Crossentropy* sebagai *loss function*. Proses pelatihan dataset model ini di jalankan sebanyak 50 *epochs* untuk kata dalam bahasa isyarat. Berikut ini merupakan visualisasi *epoch accuracy* dan *epoch loss* dari pelatihan dataset yang di dapat.



Gambar 4.13 *Epoch Accuracy*

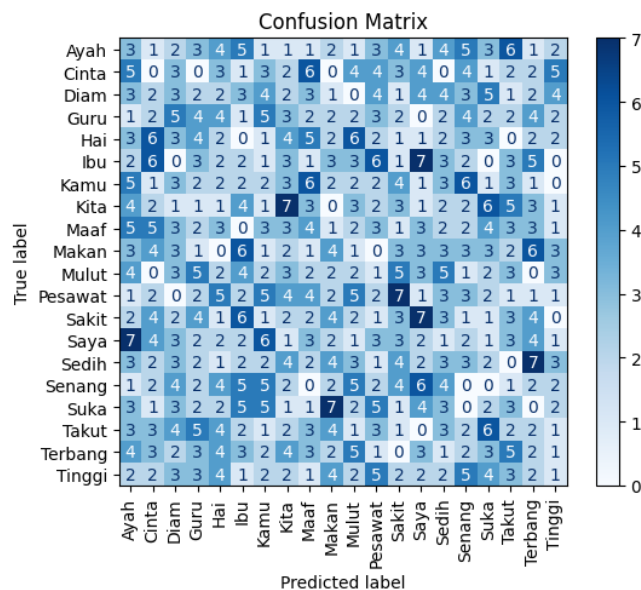
Pada Gambar 4.13 merupakan hasil visualisasi *epoch accuracy* dari pelatihan dataset kata yang mendapatkan value 0.8008 dan 0.9141 untuk validation.



Gambar 4.14 *Epoch Loss*

Pada Gambar 4.14 merupakan hasil visualisasi *epoch loss* dari pelatihan dataset kata yang mendapatkan value 0.6021 dan 0.339 untuk validation.

Kemudian model yang sudah di training di save dengan `'model.save('modeltes.h5')` yang digunakan dalam TensorFlow/Keras untuk menyimpan seluruh model ke dalam satu berkas yang komprehensif. Fungsi ini sangat berguna untuk kemudahan dalam menggunakan model yang telah dilatih di masa mendatang tanpa perlu melatih ulang. Berikut ini merupakan confusion matrix yang di dapat dari pelatihan dataset kata.

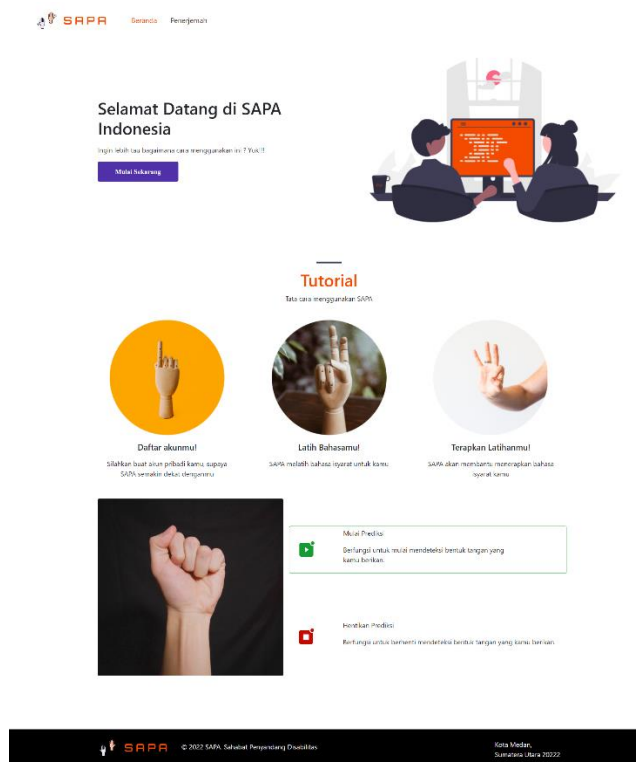


Gambar 4.15 *Confusion Matrix*

Berdasarkan hasil output confusion matrix tersebut dapat disimpulkan bahwa Matriks ini memperlihatkan jumlah prediksi yang tepat (diagonal utama) dan yang salah (elemen non-diagonal) untuk setiap kelas. Model mengklasifikasikan kelas "Ayah" dengan benar sebanyak 3 kali, namun salah mengklasifikasikannya sebagai "Cinta" sebanyak 2 kali. Selain itu, kelas "Ibu" sering kali diprediksi sebagai "Ayah" sebanyak 5 kali, menunjukkan adanya confusion antara kedua kelas tersebut.

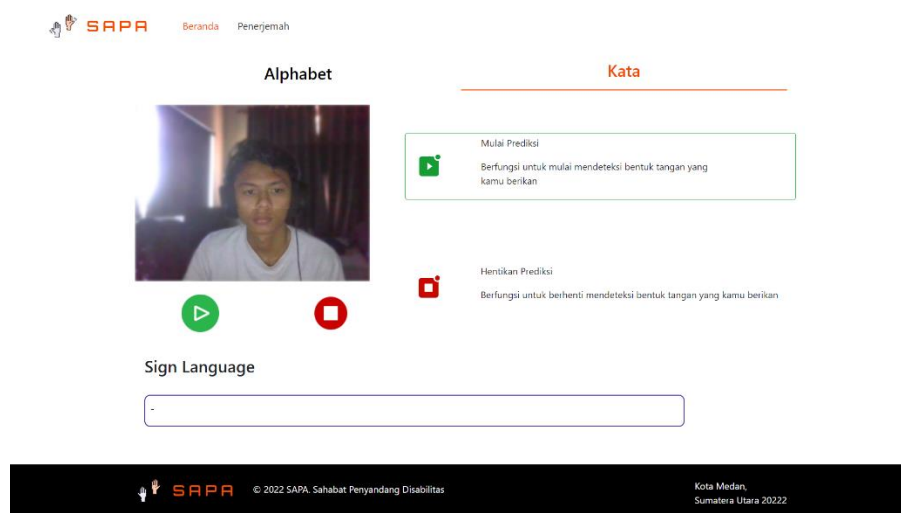
4.4.3 Hasil Pengujian

Pengujian sistem dilakukan dengan menguji sistem yang telah dibuat. Berikut ini adalah hasil pengujian sistem yang telah dilakukan. Pertama di mulai dengan menampilkan halaman Home atau Homepage dari website SAPA.



Gambar 4.16 Halaman Home

Pada halaman ini *user* mengklik pada bagian mulai penerjemah yang terdapat pada bar navigasi setelah itu user akan beralih ke bagian halaman penerjemah kata.



Gambar 4.17 Halaman Penerjemah Kata

Pada halaman penerjemah kata berfungsi untuk menerjemah gerakan menjadi sebuah kata dan terdapat penerjemah alphabet yang berfungsi untuk menerjemahkan gerakan tangan menjadi huruf.

Pada tampilan video webcam sudah mengimplementasikan *Background Blur* berikut ini adalah tampilannya ada pada gambar 4.18.










Gambar 4.18 *Background Blur*

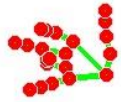


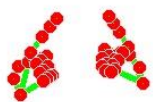


Pada halaman ini terdapat tombol start yang berfungsi untuk memulai prediksi dan tombol stop yang berfungsi untuk stop prediksi.




Pada tabel 4.1 dibawah ini merupakan 20 kosakata beserta gambar dari dataset tangan yang sudah di anotasi oleh mediapipe.

Tabel 4. 1 Kosakata dan anotasi

Kosakata	Annotasi
Ayah	
Cinta	
Diam	
Guru	
Hai	

Ibu	
Kamu	
Kita	
Maaf	
Makan	
Mulut	

Pesawat	
Sakit	
Saya	
Sedih	
Senang	
Suka	

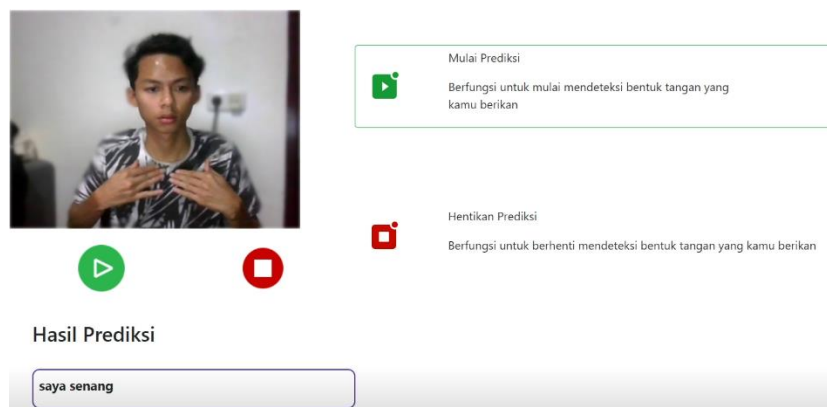
Takut	
Terbang	
Tinggi	

Untuk membuat sebuah kalimat di perlukan list kalimat dari javascript menggunakan '*const kalimat*' list kalimat tersebut dapat dilihat pada tabel 4.2 dibawah ini.

Tabel 4. 2 List Kalimat

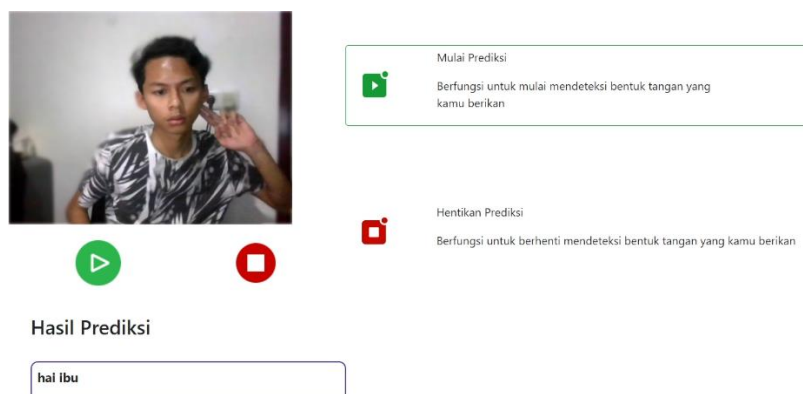
No	Kalimat
1	Saya Senang
2	Hai ibu
3	Pesawat Terbang
4	Kita sedih
5	Maaf ibu
6	Ayah Sakit
7	Diam mulut kamu

Berdasarkan Tabel 4.2 diatas terdapat 7 kalimat yang dapat di terjemahkan oleh sistem secara tepat dengan cara menggabungkan 2 gerakan bahasa isyarat. Berikut ini merupakan output dari 7 kalimat tersebut.



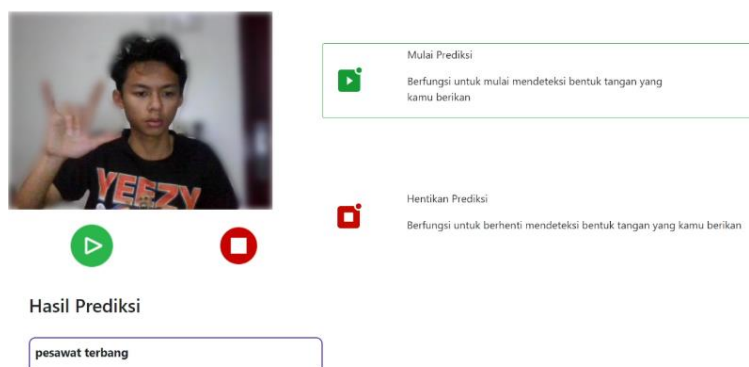
Gambar 4.19 "Saya Senang"

Pada Gambar 4.19 merupakan gerakan bahasa isyarat dari kata "Saya" dan "Senang" sehingga membentuk sebuah kalimat.



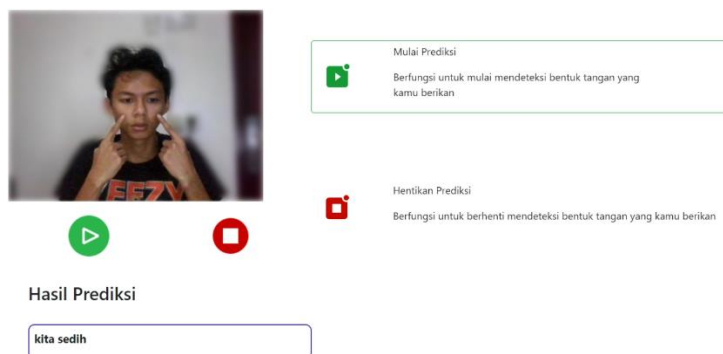
Gambar 4.20 "Hai Ibu"

Pada Gambar 4.20 merupakan gerakan bahasa isyarat dari kata "Hai" dan "Ibu" sehingga membentuk sebuah kalimat "Saya Senang".



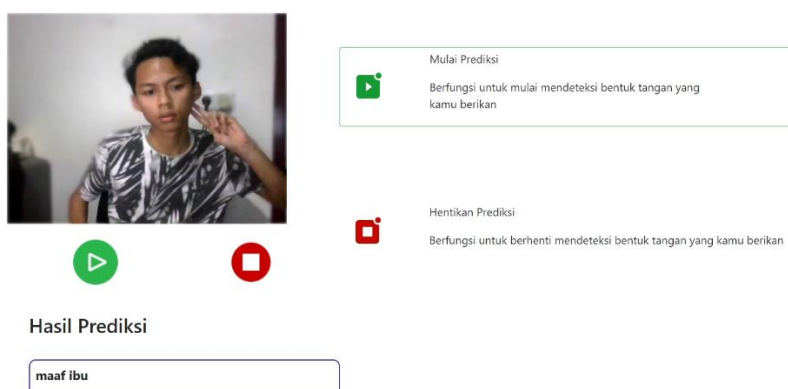
Gambar 4.21 "Pesawat Terbang"

Pada Gambar 4.21 merupakan gerakan bahasa isyarat dari kata "Pesawat" dan "Terbang" sehingga membentuk sebuah kalimat "Pesawat Terbang".



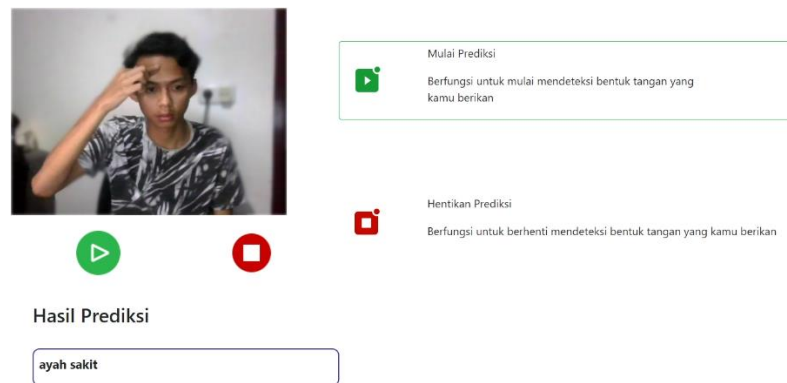
Gambar 4.22 "Kita Sedih"

Pada Gambar 4.22 merupakan gerakan bahasa isyarat dari kata "Kita" dan "Sedih" sehingga membentuk sebuah kalimat "Kita Sedih".



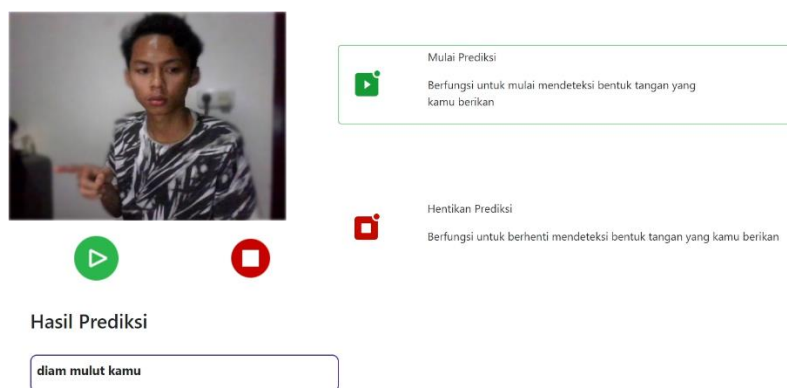
Gambar 4.23 "Maaf Ibu"

Pada Gambar 4.23 merupakan gerakan bahasa isyarat dari kata "Maaf" dan "Ibu" sehingga membentuk sebuah kalimat "Maaf Ibu".



Gambar 4.24 "Ayah Sakit"

Pada Gambar 4.24 merupakan gerakan bahasa isyarat dari kata "Ayah" dan "Sakit" sehingga membentuk sebuah kalimat "Ayah Sakit".



Gambar 4.25 "Diam Mulut Kamu"

Pada Gambar 4.25 merupakan gerakan bahasa isyarat dari kata "Diam", "Mulut" dan "Kamu" sehingga membentuk sebuah kalimat "Diam Mulut Kamu".

Gambar diatas merupakan hasil pengujian dari sistem yang sudah menerapkan algoritma *Gaussian Blur* dan berhasil mendeteksi gerakan bahasa isyarat dan menerjemahkannya kedalam bentuk kata atau kalimat. Berdasarkan pengujian sistem diatas dapat disimpulkan bahwa algoritma *Gaussian Blur* berhasil mengoptimalkan kualitas gambar dan mengurangi *noise* sehingga gerakan bahasa isyarat dapat lebih jelas teridentifikasi.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan tahap implementasi dan pengujian yang telah dilakukan pada penelitian Penerapan Algoritma *Gaussian Blur* dalam Sistem Penerjemah Bahasa Isyarat dan disimpulkan bahwa:

1. Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa penggunaan Algoritma Gaussian Blur secara efektif meningkatkan kualitas visual dari gambar sehingga dapat memudahkan dalam mendeteksi gerakan dalam sistem penerjemah bahasa isyarat.
2. Algoritma ini juga berhasil mengurangi distraksi visual dan mempertegas detail penting seperti gerakan tangan dan ekspresi wajah untuk penerjemahan bahasa isyarat.
3. Hasil pengujian dari dataset mendapatkan peningkatan akurasi, penelitian sebelumnya mendapatkan nilai akurasi 75% pada penelitian ini meningkat menjadi 91% .

5.2 Saran

Berikut adalah beberapa saran untuk penelitian selanjutnya:

1. Melakukan modifikasi dan pengujian pada variasi parameter *Gaussian Blur* yang lebih luas.
2. Mengintegrasikan *Gaussian Blur* dengan metode pengolahan gambar lain untuk memperbaiki keakuratan dan kejelasan dalam mendeteksi gerakan bahasa isyarat.
3. Mengimplementasikan *Adaptive Learning* yang dapat menyesuaikan secara dinamis parameter *Gaussian Blur* sesuai dengan kondisi video yang bervariasi untuk meningkatkan efektivitas sistem dalam berbagai kondisi penggunaan.
4. Melakukan penggabungan pada algoritma Convolutional Neural Network (CNN) dengan algoritma Long Short Term Memory (LSTM) agar dapat membuat kalimat sequence to sequence tanpa memerlukan list kalimat.

DAFTAR PUSTAKA

- Adithya, V., & Rajesh, R. (2020). A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition. *Procedia Computer Science*, 171, 2353–2361. <https://doi.org/10.1016/j.procs.2020.04.255>
- Akhmad, F., Ediansyah, P., Fitriah, J., Faramaida, E., & Purwanto, J. (2021). Karakteristik Dan Model Bimbingan atau Pendidikan Islam Bagi Abk Tuna Wicara. In *MASALIQ: Jurnal Pendidikan dan Sains* (Vol. 1, Issue 3). <https://ejournal.yasin-sys.org/index.php/masaliq>
- Aljabar, A., & Suharjito, S. (2020). BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using CNN and LSTM. *Advances in Science, Technology and Engineering Systems Journal*, 5(5), 282–287. <https://doi.org/10.25046/aj050535>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Anggraini, A., & Zakaria, H. (2023). Penerapan Metode Deep Learning Pada Aplikasi Pembelajaran Menggunakan Sistem Isyarat Bahasa Indonesia Menggunakan Convolutional Neural Network (Studi Kasus: SLB-BC Mahardika Depok). *JURIHUM: Jurnal Inovasi Dan Humaniora*, 1(4). <https://jurnalmahasiswa.com/index.php/jurihum>
- Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. *Clinical Pharmacology and Therapeutics*, 107(4), 871–885. <https://doi.org/10.1002/cpt.1796>
- Bailey, P., Bouaziz, S., Carter, S., Gordon, J., Häne, C., Mordvintsev, A., Valentin, J., & Wicke, M. (2019, July 28). Differentiable graphics with TensorFlow 2.0. ACM SIGGRAPH 2019 Courses, SIGGRAPH 2019. <https://doi.org/10.1145/3305366.3328041>
- Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378, 112–119. <https://doi.org/10.1016/j.neucom.2019.10.008>
- Cai, S., Shu, Y., Chen, G., Ooi, B. C., Wang, W., & Zhang, M. (2019). *Effective and Efficient Dropout for Deep Convolutional Neural Networks*. <http://arxiv.org/abs/1904.03392>
- Cao, J., Li, Y., Sun, M., Chen, Y., Lischinski, D., Cohen-Or, D., Chen, B., & Tu, C. (2022). DO-Conv: Depthwise Over-Parameterized Convolutional Layer. *IEEE Transactions on Image Processing*, 31, 3726–3736. <https://doi.org/10.1109/TIP.2022.3175432>

- Ce, P., & Tie, B. (2020). An analysis method for interpretability of CNN text classification model. *Future Internet*, 12(12), 1–14. <https://doi.org/10.3390/fi12120228>
- Chandwani, L., Khilari, J., Gurjar, K., Maragale, P., Sonare, A., Kakade, S., Bhatt, A., & Kulkarni, R. (2023). *Gesture based Sign Language Recognition system using Mediapipe*. <https://doi.org/10.21203/rs.3.rs-3106646/v1>
- Christlein, V., Spranger, L., Seuret, M., Nicolaou, A., Král, P., & Maier, A. (2019). *Deep Generalized Max Pooling*. <http://arxiv.org/abs/1908.05040>
- Das, P., Ahmed, T., & Ali, Md. F. (2020). Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network. *2020 IEEE Region 10 Symposium (TENSYP)*, 1762–1765. <https://doi.org/10.1109/TENSYP50017.2020.9230772>
- Devi, T. G., Patil, N., Rai, S., & Philipose, C. S. (2023). Gaussian Blurring Technique for Detecting and Classifying Acute Lymphoblastic Leukemia Cancer Cells from Microscopic Biopsy Images. *Life*, 13(2). <https://doi.org/10.3390/life13020348>
- Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., Zhang, C., & Huang, W. (2020). On line detection of defective apples using computer vision system combined with deep learning methods. *Journal of Food Engineering*, 286. <https://doi.org/10.1016/j.jfoodeng.2020.110102>
- Fatmawati, R., Asmara, R., Prayogi, Y. R., & Hakkun, R. Y. (2022). Aplikasi Pembelajaran Sistem Isyarat Bahasa Indonesia (SIBI) Berbasis Voice Menggunakan OpenSIBI. *Technomedia Journal (TMJ)*, 7(1), 22–39. <https://doi.org/10.33050/tmj.v7i1%20Juni.1690>
- Hassan, H., Bashir, A. K., Abbasi, R., Ahmad, W., & Luo, B. (2019). Single image defocus estimation by modified gaussian function. *Transactions on Emerging Telecommunications Technologies*, 30(6). <https://doi.org/10.1002/ett.3611>
- Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, 9(4). <https://doi.org/10.1002/widm.1312>
- Huang, G., Tran, S. N., Bai, Q., & Alty, J. (2023). Real-time automated detection of older adults' hand gestures in home and clinical settings. *Neural Computing and Applications*, 35(11), 8143–8156. <https://doi.org/10.1007/s00521-022-08090-8>
- Ignatov, A., Patel, J., & Timofte, R. (2020). Rendering Natural Camera Bokeh Effect with Deep Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020-June, 1676–1686. <https://doi.org/10.1109/CVPRW50498.2020.00217>
- Iskandar, R., & Supena, A. (2021). Implementasi Layanan Inklusi Anak Berkebutuhan Khusus Tunarungu. *Jurnal Komunikasi Pendidikan*, 5(1), 124. <https://doi.org/10.32585/jkp.v5i1.1018>

- Jeczminek, E., & Kowalski, P. A. (2021). Flattening layer pruning in convolutional neural networks. *Symmetry*, 13(7). <https://doi.org/10.3390/sym13071147>
- Juniar, A. N., Mahdi, S., Hendra, H., Wardhani, G. L., & Raudah, R. (2023). Sistem Otomatisasi blur citra Wajah Menggunakan gaussian filter Dan DLIB CNN. *Elektrika Borneo*, 9(2). <https://doi.org/10.35334/eb.v3i2.2307>
- Li, H., Liu, C., & Basu, A. (2022). Semantic Segmentation Based on Depth Background Blur. *Applied Sciences (Switzerland)*, 12(3). <https://doi.org/10.3390/app12031051>
- Natarajan, B., Rajalakshmi, E., Elakkiya, R., Kotecha, K., Abraham, A., Gabralla, L. A., & Subramaniaswamy, V. (2022). Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation. *IEEE Access*, 10, 104358–104374. <https://doi.org/10.1109/ACCESS.2022.3210543>
- Naufal, M. F., & Kusuma, S. F. (2023). Analisis Perbandingan Algoritma Machine Learning dan Deep Learning untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(4), 873–882. <https://doi.org/10.25126/jtiik.20241046823>
- Nur, M., Mannessa, H. I. S., Muchsin, N. R., & Andranatha, A. A. (2022). Pelatihan Bahasa Isyarat Di Lingkup Sentra Wirajaya di Makassar Dalam Membangun Komunikasi Efektif Kepada Tunarungu (Tuli) Irdianti Fakultas Psikologi Universitas Negeri Makassar, Indonesia Irdiantipsi@unm.ac.id. *Journal of Community Dedication*, 2(4), 196–210.
- Sabeenian, R. S., Sai, B. S., & Mohamed, A. M. (2020). Sign Language Recognition Using Deep Learning and Computer Vision. *Journal of Advanced Research in Dynamical and Control Systems*, 12(05-SPECIAL ISSUE), 964–968. <https://doi.org/10.5373/JARDCS/V12SP5/20201842>
- Shah, F., Shah, M. S., Akram, W., Manzoor, A., Mahmoud, R. O., & Abdelminaam, D. S. (2021). Sign Language Recognition Using Multiple Kernel Learning: A Case Study of Pakistan Sign Language. *IEEE Access*, 9, 67548–67558. <https://doi.org/10.1109/ACCESS.2021.3077386>
- Shen, Y., Xu, W., Lin, Q., Allebach, J. P., & Zhu, F. (2023). Depth Assisted Portrait Video Background Blurring. *IS and T International Symposium on Electronic Imaging Science and Technology*, 35(7). <https://doi.org/10.2352/EI.2023.35.7.IMAGE-273>
- Sholawati, M., Auliasari, K., & Ariwibisono, F. X. (2022). Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn). In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Issue 1).
- Shreya, D. S. (2021). Digital Image Processing and Recognition Using Python. In *International Journal of Engineering Applied Sciences and Technology* (Vol. 5). <http://www.ijeast.com>
- Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S. N.,

- Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F. B., & Wattenberg, M. (2019). Tensorflow.js: Machine Learning For The Web And Beyond.
- Suharjito, Thiracitta, N., & Gunawan, H. (2021). SIBI Sign Language Recognition Using Convolutional Neural Network Combined with Transfer Learning and non-trainable Parameters. *Procedia Computer Science*, 179, 72–80. <https://doi.org/10.1016/j.procs.2020.12.011>
- Tabellion, E., Karnad, N., Glaser, N., Weiss, B., Jacobs, D. E., & Pritch, Y. (2023). Computational Long Exposure Mobile Photography. *ACM Transactions on Graphics*, 42(4), 1–15. <https://doi.org/10.1145/3592124>
- Vakalopoulou, M., Christodoulidis, S., Burgos, N., Colliot, O., & Lepetit, V. (2023). Deep Learning: Basics and Convolutional Neural Networks (CNNs). In *Neuromethods* (Vol. 197, pp. 77–115). Humana Press Inc. https://doi.org/10.1007/978-1-0716-3195-9_3
- Vaz, J. M., & Balaji, S. (2021). Convolutional neural networks (CNNs): concepts and applications in pharmacogenomics. *Molecular Diversity*, 25(3), 1569–1584. <https://doi.org/10.1007/s11030-021-10225-3>
- WHO. (2021). *Global report on health equity for persons with disabilities*.
- Xu, Y., Liu, X., Cao, X., Huang, C., Liu, E., Qian, S., Liu, X., Wu, Y., Dong, F., Qiu, C. W., Qiu, J., Hua, K., Su, W., Wu, J., Xu, H., Han, Y., Fu, C., Yin, Z., Liu, M., ... Zhang, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. In *Innovation* (Vol. 2, Issue 4). Cell Press. <https://doi.org/10.1016/j.xinn.2021.100179>
- Zhu, Q., He, Z., Zhang, T., & Cui, W. (2020). Improving classification performance of softmax loss function based on scalable batch-normalization. *Applied Sciences (Switzerland)*, 10(8). <https://doi.org/10.3390/APP10082950>
- Patel, K. (2019). MNIST Handwritten Digits Classification using a Convolutional Neural Network (CNN). Towards Data Science. <https://towardsdatascience.com>