

**PERANCANGAN APLIKASI PENERJEMAH BAHASA
ISYARAT MENGGUNAKAN METODE *HAND
GESTURE RECOGNITION* & ALGORITMA
*RANDOM FOREST CLASSIFIER***

SKRIPSI

Oleh:

ANDRIAN ANJAYA

NIM 2044026



**JENJANG PENDIDIKAN STRATA-1
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME
MEDAN
2024**

LEMBAR PENGESAHAN

**PERANCANGAN APLIKASI PENERJEMAH BAHASA
ISYARAT MENGGUNAKAN METODE *HAND
GESTURE RECOGNITION & ALGORITMA
RANDOM FOREST CLASSIFIER***

SKRIPSI

Diajukan Untuk Melengkapi Persyaratan Guna
Mendapat Gelar Sarjana Strata Satu
Program Studi Teknik Informatika

Oleh:

**ANDRIAN ANJAYA
NIM 2044026**

Disetujui Oleh:

Dosen Pembimbing I



Robet, M.Kom

Dosen Pembimbing II



Chandra, M.Kom

Medan, 26 Juli 2024

Diketahui dan Disahkan Oleh:

Ketua STMIK TIME

Ketua Program Studi



Dr. Edi Wijaya, S.Kom., M.Kom., M.M

Robet, M.Kom

Kepada
Yth Ketua STMIK TIME
Dr. Edi Wijaya, S.Kom., M.Kom., M.M
di Medan

Hal : **Surat Pernyataan Bebas Plagiat**

Saya yang bertanda tangan di bawah ini :

Nama : Andrian Anjaya
NIM : 2044026
Program Studi : Teknik Informatika
Angkatan : 2020
Judul Skripsi : Perancangan Aplikasi Penerjemah Bahasa Isyarat
Menggunakan Metode *Hand Gesture Recognition*
& Algoritma *Random Forest Classifier*

Dengan ini menyatakan bahwa skripsi yang Saya tulis adalah karya ilmiah asli dari pikiran saya sendiri.

Apabila kemudian hari terbukti bahwa skripsi tersebut merupakan Plagiat, maka saya bersedia gelar dan ijazah yang saya peroleh ditinjau kembali.

Demikian surat pernyataan ini Saya perbuat dengan pikiran yang sehat, tanpa tekanan dari siapapun.

Diketahui oleh
Pembantu Ketua I



Hendri, M.Kom

Medan, 30 Juli 2024
Yang membuat pernyataan



Andrian Anjaya

ABSTRAK

Penelitian ini mengembangkan aplikasi penerjemah bahasa isyarat berbasis *web* yang bertujuan untuk memudahkan komunikasi bagi penyandang disabilitas tunarungu. Aplikasi ini menggunakan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* untuk mendeteksi dan menerjemahkan *gesture* tangan ke dalam teks secara *real-time*. Dalam penelitian ini, dataset lokal digunakan dengan setiap kata bahasa isyarat memiliki 100 gambar data latih, dan *gesture* yang diterjemahkan dibatasi pada huruf alphabet A hingga Z serta beberapa kata umum seperti "*hello*", "*done*", "*thank you*", "*I love you*", "*sorry*", "*please*", dan "*you are welcome*". Hasil penelitian menunjukkan bahwa aplikasi yang dibangun berhasil mencapai akurasi 84% dalam menerjemahkan *gesture* bahasa isyarat ke dalam teks, membuktikan bahwa metode dan algoritma yang digunakan efektif dalam mendeteksi dan mengklasifikasikan gestur tangan. Aplikasi ini diharapkan dapat membantu penyandang tunarungu berkomunikasi lebih efektif, meningkatkan aksesibilitas informasi, dan berkontribusi pada pengembangan teknologi informatika di bidang pengenalan gestur tangan dan penerjemahan bahasa isyarat.

Kata Kunci: Aplikasi, Penerjemah Bahasa Isyarat, Metode *Hand Gesture Recognition*, Algoritma *Random Forest Classifier*

ABSTRACT

This research developed a web-based sign language translation application aimed at facilitating communication for individuals with hearing disabilities. The application uses the Hand Gesture Recognition method and the Random Forest Classifier algorithm to detect and translate hand gestures into text in real-time. In this study, a local dataset was used, with each sign language word having 100 training data images, and the gestures translated were limited to the alphabet letters A to Z and several common words such as "hello," "done," "thank you," "I love you," "sorry," "please," and "you are welcome." The research results indicate that the developed application successfully achieved an 84% accuracy rate in translating sign language gestures into text, demonstrating that the methods and algorithms used are effective in detecting and classifying hand gestures. This application is expected to help individuals with hearing disabilities communicate more effectively, enhance information accessibility, and contribute to the development of information technology in the field of hand gesture recognition and sign language translation.

Keywords: Application, Sign Language Translator, Hand Gesture Recognition Method, Random Forest Classifier Algorithm

KATA PENGANTAR

Ucapan puji dan syukur kepada hadirat Tuhan Yang Maha Esa atas segala berkat, hikmat, dan pertolongan-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini tepat pada waktunya.

Tugas Akhir ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Program Strata I pada STMIK TIME. Penulis berharap agar hasil dari penelitian yang dilakukan ini dapat membawa banyak manfaat bagi berbagai pihak.

Dalam menyelesaikan tugas akhir ini, penulis banyak mendapat bantuan dari berbagai pihak. Oleh karena itu, melalui kesempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Bapak Robet, M.Kom., selaku Pembimbing I dan Ketua Program Studi Teknik Informatika STMIK TIME Medan yang telah membimbing penulis selama mengerjakan skripsi ini.
2. Bapak Chandra, M.Kom., selaku Pembimbing II yang telah membimbing penulis selama mengerjakan skripsi ini.
3. Bapak Simon Kangga Lee, selaku Ketua Yayasan STMIK TIME Medan.
4. Bapak Dr. Edi Wijaya, S.Kom., M.Kom., M.M, selaku Ketua STMIK TIME Medan yang telah membimbing penulis selama mengerjakan skripsi ini.
5. Bapak Hendri, M.Kom selaku Pembantu Ketua I STMIK TIME Medan yang telah membimbing penulis selama mengerjakan skripsi ini.
6. Seluruh Dosen STMIK TIME Medan yang telah mendidik dan membimbing penulis selama masa perkuliahan.
7. Teristimewa kepada orang tua dan keluarga tercinta, yang telah memberikan dukungan baik moral maupun materi kepada penulis selama ini.
8. Seluruh teman yang telah banyak membantu penulis dan memberikan dorongan serta semangat dalam menyelesaikan skripsi ini
9. Semua yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak dapat disebutkan satu persatu

Penulis menyadari dalam proses penyelesaian skripsi yang berjudul “Perancangan Aplikasi Penerjemah Bahasa Isyarat Menggunakan Metode *Hand Gesture Recognition* & Algoritma *Random Forest Classifier*” masih jauh dari sempurna, mengingat keterbatasan kemampuan yang penulis miliki dan terbatasnya waktu yang digunakan. Untuk itu, penulis mengharapkan saran, kritikan, dan masukan yang membangun demi perkembangan dan kemajuan penelitian selanjutnya. Semoga penulisan skripsi ini dapat bermanfaat bagi pihak yang berkepentingan.

Akhir kata penulis berharap agar laporan penulisan Tugas Akhir ini dapat berguna dan bermanfaat bagi pembaca.

Medan, 30 Juli 2024

Penulis,

Andrian Anjaya

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
LAMPIRAN	Error! Bookmark not defined.
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	Error! Bookmark not defined.
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat Penelitian	4
1.5 Metodologi Penelitian	5
1.6 Analisis Sistem.....	6
1.7 Perancangan Sistem.....	7
1.8 Sistematika Penulisan.....	7
BAB II TINJAUAN PUSTAKA	9
2.1 Digital Image	9
2.2 Digital Image Processing	12
2.3 Computer Vision	15
2.4 Kecerdasan Buatan.....	18
2.5 Machine Learning	21
2.6 Knowledge Discovery	23
2.7 Metode Hand Gesture Recognition.....	26
2.8 Convolutional Neural Network (CNN)	27
2.9 Algoritma Random Forest Classifier	30
2.10 Python.....	31

2.11 Penelitian Terdahulu	32
BAB III METODE PENELITIAN.....	34
3.1 Analisis Masalah.....	34
3.1.1 Analisis Proses Yang Berjalan Sesuai di Lapangan.....	34
3.1.2 Analisis Metode dan Algoritma Yang Digunakan.....	35
3.1.3 Analisis Sistem Usulan.....	42
3.2 Perancangan.....	45
BAB IV HASIL DAN PEMBAHASAN.....	51
4.1 Hasil	51
4.2 Pembahasan	54
BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	63
DAFTAR PUSTAKA.....	65

DAFTAR GAMBAR

Gambar 2.1 Representasi Citra RGB dan Kanal Warna Penyusunnya [6]	9
Gambar 2.2 Representasi piksel dengan kombinasi warna R, G, dan B [6]	10
Gambar 2.3 Citra Hasil Konversi RGB Menjadi <i>Grayscale</i> [6]	11
Gambar 2.4 Citra Hasil Konversi RGB Menjadi Biner [6].....	12
Gambar 2.5 Ilustrasi Komputer Melihat Citra [7]	17
Gambar 2.6 Perbedaan <i>Supervised Learning</i> dan <i>Unsupervised Learning</i> [10] ...	23
Gambar 2.7 Proses <i>Knowledge Discovery</i> Pada <i>Database</i> [9].....	24
Gambar 2.8 Arsitektur <i>Convolutional Neural Network</i> [13]	27
Gambar 2.9 Ilustrasi Arsitektur <i>Convolutional Neural Network</i> [13]	27
Gambar 2.10 Contoh Operasi Matematis Proses Konvolusi [13]	28
Gambar 2.11 Contoh Penerapan Aktivasi ReLU [13]	29
Gambar 2.12 IDLE Python [7]	32
Gambar 3.1 <i>Flowchart</i> Alir Penelitian.....	36
Gambar 3.2 Contoh <i>Dataset</i> Penelitian.....	37
Gambar 3.3 <i>Dataset</i> Bahasa Isyarat Makan	37
Gambar 3.4 Citra <i>Grayscale</i>	38
Gambar 3.5 <i>Use Case Diagram</i> Aplikasi Penerjemah Bahasa Isyarat	42
Gambar 3.6 Rancangan Halaman Awal	46
Gambar 3.7 Rancangan Halaman <i>Collect and Create Dataset</i>	47
Gambar 3.8 Rancangan Halaman <i>Collect Dataset Using Webcam</i>	48
Gambar 3.9 Rancangan Halaman <i>Training Dataset</i>	49
Gambar 3.10 Rancangan Halaman Penerjemahan Bahasa Isyarat	50
Gambar 4.1 Halaman Awal	51
Gambar 4.2 Halaman <i>Collect and Create Dataset</i>	52
Gambar 4.3 Halaman <i>Collect Dataset Using Webcam</i>	52
Gambar 4.4 Halaman <i>Training Dataset</i>	53
Gambar 4.5 Halaman Penerjemahan Bahasa Isyarat	54
Gambar 4.6 Grafik Hasil Pengujian.....	59

DAFTAR TABEL

Tabel 1.1 Jadwal Penelitian.....	Error! Bookmark not defined.
Tabel 3.1 Skenario <i>Use Case</i> Mengumpulkan Dataset.....	43
Tabel 3.2 Skenario <i>Use Case</i> Membuat Dataset	43
Tabel 3.3 Skenario <i>Use Case</i> Melakukan <i>Training</i> Dataset	44
Tabel 3.4 Skenario <i>Use Case</i> Mencoba Penerjemah Bahasa Isyarat.....	45
Tabel 4.1 Hasil Pengujian	55
Tabel 4.2 Hasil Pengujian <i>Epoch</i>	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tunarungu atau disabilitas pendengaran adalah seseorang yang mengalami gangguan pendengaran, baik sebagian ataupun menyeluruh, dan biasanya memiliki hambatan dalam berbahasa dan berbicara. Tunarungu memiliki keterbatasan dalam indra sehingga berdampak pada kehidupannya sehari-hari terutama dalam kemampuan berbahasa yang digunakan sebagai sarana komunikasi penting untuk bersosialisasi dengan orang lain [1]. Oleh sebab itu, penyandang tunarungu menggunakan bahasa isyarat untuk berkomunikasi dengan individu lainnya [2]. SIBI atau Sistem Isyarat Bahasa Indonesia merupakan bahasa yang digunakan oleh penyandang tunarungu sebagai media untuk berkomunikasi dengan masyarakat [3].

Namun, terdapat kesulitan dalam berkomunikasi tunarungu dengan orang-orang yang tidak mengerti bahasa isyarat sehingga diperlukan teknologi informasi yang dapat membantu menerjemahkan bahasa isyarat kedalam bahasa lisan sehingga memudahkan komunikasi bagi kedua belah pihak [3]. Penelitian yang membahas mengenai implementasi teknologi informasi dalam penerjemahan bahasa isyarat sudah pernah dilakukan sebelumnya yaitu pada tahun 2022 yang membahas mengenai perancangan aplikasi penerjemah bahasa isyarat ke dalam huruf abjad menggunakan algoritma klasifikasi *Convolutional Neural Network* (CNN). Hasil penelitian tersebut menyimpulkan bahwa algoritma CNN dalam menerjemahkan bahasa isyarat mendapatkan akurasi 0,995 atau 99,5% [4].

Penelitian terbaru yaitu di tahun 2023 mengimplementasikan algoritma *Long-Short Term Memory* (LSTM) dalam pengembangan aplikasi penerjemah bahasa isyarat. Hasil penelitian menunjukkan bahwa algoritma LSTM mendapatkan akurasi di antara 75%-85% [3].

Pada penelitian ini akan dikembangkan aplikasi penerjemah bahasa isyarat secara *real-time* dengan menggunakan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier*. Metode *Hand Gesture Recognition* memiliki kelebihan dalam mendeteksi gestur tangan dengan akurasi tinggi dan kemampuan adaptasi yang baik terhadap variasi gestur tangan sehingga hasil gerakan isyarat jari terdeteksi dengan akurat. Setelah gestur tangan terdeteksi, maka diimplementasikan algoritma *Random Forest Classifier* untuk klasifikasi gestur tersebut ke dalam bahasa isyarat yang relevan sesuai dengan data pelatihan yang ada, sehingga gestur bahasa isyarat yang dibuat dapat diterjemahkan ke dalam teks sehingga memudahkan komunikasi. Penelitian ini memilih algoritma *Random Forest Classifier* dikarenakan berdasarkan penelitian terdahulu algoritma ini memiliki kelebihan dalam hal kecepatan komputasi, kemampuan pemrosesan data yang lebih baik serta tingkat akurasi yang stabil dibandingkan algoritma CNN dan LSTM [5][6].

Berdasarkan uraian sebelumnya, dilakukan penelitian dengan mengambil judul **“Perancangan Aplikasi Penerjemah Bahasa Isyarat Menggunakan Metode *Hand Gesture Recognition* dan Algoritma *Random Forest Classifier*”**.

1.2 Rumusan Masalah

Sesuai uraian latar belakang masalah tersebut, dirumuskan permasalahan pada penelitian ini yaitu:

1. Bagaimana membangun sebuah aplikasi penerjemah bahasa isyarat sehingga memudahkan komunikasi para penyandang disabilitas tunarungu?
2. Bagaimana menganalisis akurasi dari metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* dalam menerjemahkan gesture bahasa isyarat ke dalam teks?

1.3 Batasan Masalah

Berikut ini akan dilakukan pembatasan masalah penelitian dengan tujuan agar penelitian dapat diselesaikan tepat pada waktu yang ditentukan antara lain:

1. *Dataset* pada penelitian ini diambil dari dataset lokal dengan setiap kata bahasa isyarat masing-masing memiliki 100 gambar data latih.
2. *Gesture* bahasa isyarat yang dapat diterjemahkan dibatasi hanya pada huruf alphabet dari A hingga Z, kata *hello, done, thank you, i love you, sorry, please,* dan *you are welcome*.
3. *Output* hasil terjemahan disajikan dalam bentuk papan teks yang dapat dilihat di *WebCam*.
4. Aplikasi yang dibangun berbasis *website* dan bersifat *real time*.

1.4 Tujuan dan Manfaat Penelitian

Tujuan yang diharapkan dalam penelitian ini adalah:

1. Untuk membangun sebuah aplikasi penerjemah bahasa isyarat sehingga memudahkan komunikasi para penyandang disabilitas tunarungu.
2. Untuk menguji dan menganalisis akurasi dari metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* dalam menerjemahkan gesture bahasa isyarat ke dalam teks.

Manfaat yang diharapkan dalam penelitian ini adalah:

1. Aplikasi penerjemah bahasa isyarat yang dikembangkan dapat membantu penyandang tunarungu berkomunikasi secara efektif dengan orang-orang yang tidak mengerti bahasa isyarat. Hal ini akan memungkinkan mereka untuk berpartisipasi dalam berbagai interaksi sosial dan pekerjaan dengan lebih mudah dan tanpa hambatan komunikasi.
2. Aplikasi yang dibangun meningkatkan aksesibilitas informasi dan komunikasi bagi penyandang tunarungu. Penyandang tunarungu dapat mengakses berbagai layanan dan informasi yang sebelumnya sulit dijangkau karena hambatan bahasa isyarat. Dengan demikian, penelitian ini akan membantu menciptakan lingkungan yang lebih inklusif bagi penyandang tunarungu.
3. Penelitian ini juga berkontribusi pada pengembangan teknologi informatika khususnya dalam bidang pengenalan gestur tangan dan penerjemahan bahasa isyarat. Penggunaan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* dalam konteks ini dapat membuka peluang untuk pengembangan teknologi serupa dalam berbagai aplikasi lainnya, seperti pengenalan gerakan tubuh atau gestur dalam berbagai konteks lainnya.

1.5 Metodologi Penelitian

Tahapan-tahapan metodologi penelitian yang akan dilaksana pada penelitian ini diuraikan dalam beberapa poin berikut:

1. Jadwal Penelitian

Jadwal penelitian berisikan informasi pelaksanaan penelitian dari awal penelitian hingga penelitian selesai yang dapat dilihat pada Tabel 1.1.

Tabel 1.1 Jadwal Penelitian

NO	Waktu	2023								2024															
		November				Desember				Januari				Februari				Maret				April			
	Tahapan	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Rumusan Masalah																								
2	Pengumpulan Data																								
3	Analisis Sistem																								
4	Perancangan Sistem																								
5	Uji Coba Sistem																								
6	Penulisan Laporan																								

2. Bahan dan Alat

Bahan dan alat penelitian dibagi menjadi 2 yaitu dari sisi *hardware* (perangkat keras) dan *software* (perangkat lunak).

Hardware yang digunakan untuk melaksanakan penelitian yaitu sebuah laptop dengan spesifikasi:

- Processor Intel® Core™ i5-8300H CPU @ 2,30GHz
- RAM 8GB.
- Perangkat *keyboard*.
- Perangkat *touchpad*.

- e. *Display* dengan resolusi 1920 x 1080 pixel.
- f. *Webcam* dengan spesifikasi resolusi 1080p/ 30 fps, *auto focus*, lensa *full HD Glass*, *field of view* (sudut pandang) 78°, dan *microphone stereo*.

Software yang digunakan untuk melaksanakan penelitian yaitu:

- a. Sistem Operasi Windows 11 64 Bit.
- b. *IDLE Python*.
- c. Python versi 3.8.
- d. *Library OpenCV* dan *Scikit-Learn*.
- e. Flask.

3. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini antara lain sebagai berikut:

a. Metode Observasi

Melakukan observasi secara langsung yaitu mendatangi objek penelitian untuk mengumpulkan data-data *gesture* jari dari beberapa responden untuk mendeteksi *gesture* bahasa isyarat kosakata.

b. Metode Studi Pustaka

Mengumpulkan data-data teori melalui jurnal, media cetak, ataupun sumber-sumber referensi dari internet.

1.6 Analisis Sistem

Analisis sistem pada penelitian ini terbagi menjadi 3 tahapan proses yaitu:

1. Analisis proses yang berjalan sesuai di lapangan yaitu sebuah pencermatan terhadap implementasi suatu kegiatan atau sistem yang berlangsung dengan

baik dan sesuai dengan rencana atau standar yang telah ditetapkan. Proses analisis sistem berjalan menguraikan secara mendalam tahapan sistem berjalan yang digunakan saat ini dalam proses menerjemahkan bahasa isyarat.

2. Analisis metode dan algoritma yang digunakan berisikan tahapan-tahapan yang berupa pengauran cara kerja dari metode *Hand Gesture Recognition* dalam mendeteksi *gesture* jari dan cara kerja serta perhitungan manual dari algoritma *Random Forest Classifier* dalam melakukan klasifikasi terhadap data-data yang ada sebelumnya untuk menerjemahkan bahasa isyarat.
3. Analisis sistem usulan yaitu menggambarkan sistem usulan yang akan dibangun beserta fitur-fitur yang tersedia yang dimodelkan dengan menggunakan *tools* pemodelan sistem *Use Case Diagram*.

1.7 Perancangan Sistem

Tahapan perancangan sistem yang dilakukan pada penelitian ini adalah berupa perancangan tampilan aplikasi dengan menggunakan *software* Balsamiq Mockup 3.

1.8 Sistematika Penulisan

Untuk mempermudah memahami permasalahan yang dibahas, maka skripsi ini akan disusun dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bagian ini berisikan informasi awal dari penelitian yang akan dilaksanakan yang terdiri dari latar belakang, rumusan masalah,

batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bagian ini dibahas mengenai landasan dan kajian teori-teori yang berhubungan dengan judul penulisan skripsi serta penelitian terdahulu yang pernah dilakukan sebelumnya untuk mencari *research gap*.

BAB III METODE PENELITIAN

Bagian ini membahas mengenai tahapan metode penelitian yang dilakukan pada penelitian. Pada bab ini dibagi menjadi 2 pembahasan penting yaitu analisis dan perancangan.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menguraikan tahapan yang penulis lakukan dalam mengembangkan aplikasi dan hasil implementasi yang didapatkan.

BAB V KESIMPULAN DAN SARAN

Bagian ini berisikan mengenai kesimpulan yang didapatkan setelah melakukan penelitian dengan proses analisa terhadap teori dan hasil pengujian. Selain itu juga diberikan saran yang berisikan hal-hal yang belum berhasil diselesaikan sehingga dapat dilanjutkan pada penelitian berikutnya.

BAB II

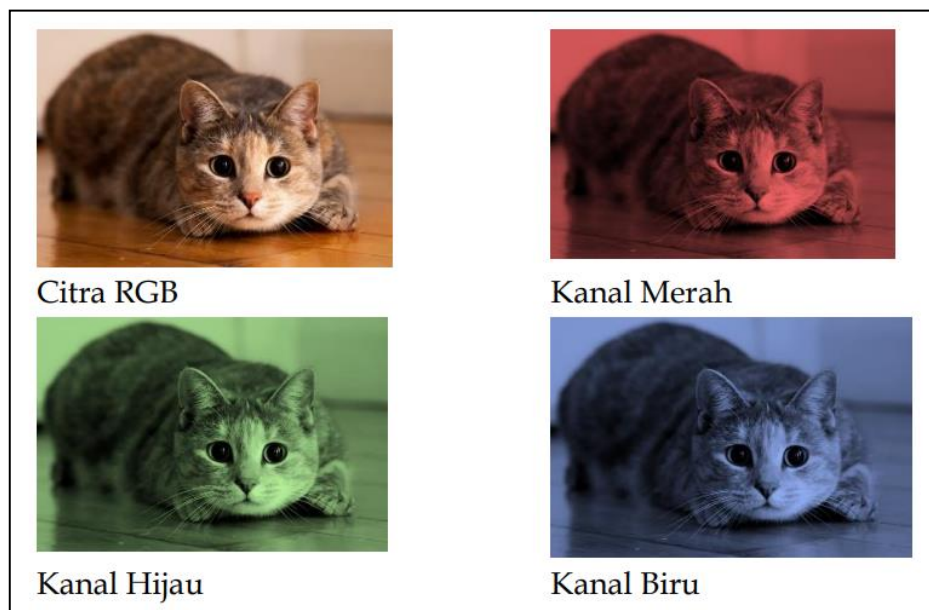
TINJAUAN PUSTAKA

2.1 Digital Image

Citra digital atau digital *image* adalah gambar yang terdiri dari elemen gambar, juga dikenal sebagai piksel, masing-masing dengan jumlah representasi numerik yang terbatas dan diskrit untuk intensitas atau tingkat keabuannya [7]. Kombinasi warna pada sebuah piksel dalam citra dapat diuraikan menjadi tiga bagian antara lain [7]:

1. Citra RGB

Citra RGB ini disusun oleh tiga warna utama yaitu warna merah, hijau dan biru. Pada warna ini, masing-masing memiliki nilai intensitas pada pikselnya yang masing-masing memiliki 8-bit atau 2^8 atau 256 variasi warna.



Gambar 2.1 Representasi Citra RGB dan Kanal Warna Penyusunnya [6]

Kombinasi warna dalam sebuah piksel yang mungkin dapat terbentuk pada citra RGB yaitu sebanyak $256^3 = 16.777.216$. Pada Gambar 2.2 dapat dilihat contoh-contoh kombinasi warna yang dapat terbentuk dengan nilai RGB yang berbeda.

Yellow R = 255 G = 255 B = 0	Orange R = 255 G = 102 B = 0	Green R = 0 G = 255 B = 0
Cyan R = 0 G = 255 B = 255	Violet R = 204 G = 102 B = 204	White R = 255 G = 255 B = 255
Black R = 0 G = 0 B = 0	Turquoise R = 102 G = 255 B = 204	Brown R = 153 G = 102 B = 51

Gambar 2.2 Representasi piksel dengan kombinasi warna R, G, dan B [6]

2. Citra *Grayscale*

Citra *grayscale* adalah citra yang dihitung nilai intensitas pada pikselnya berdasarkan tingkat keabuan. Dalam sebuah citra *grayscale* yang memiliki nilai 8-bit, tingkat keabuan tersebut dibagi menjadi 256 tingkat keabuan dimana nilai 255 merupakan putih total dan nilai 0 merupakan hitam total. Sebuah citra RGB dapat dibuat menjadi satu kanal warna dengan cara mengkonversikan citra tersebut menjadi citra *grayscale*. Cara mengkonversi citra RGB menjadi citra *grayscale* yaitu dengan menggunakan formula:

$$Grayscale = 0,2989 * R + 0,5870 * G + 0,1140 * B$$

dimana:

- Grayscale* merupakan nilai intensitas citra *grayscale*.

- b. R merupakan nilai intensitas merah pada piksel.
- c. G adalah nilai intensitas hijau pada piksel.
- d. B adalah nilai intensitas biru pada piksel.

Pada Gambar 2.3 terdapat hasil konversi Gambar 2.1 yang merupakan citra RGB menjadi citra *grayscale*.



Gambar 2.3 Citra Hasil Konversi RGB Menjadi *Grayscale* [6]

3. Citra Biner

Citra biner ini hanya memiliki dua nilai intensitas dan nilai bit sebesar 1. Oleh karena itu citra ini memiliki dua nilai intensitas yaitu warna hitam ataupun 0 dan warna putih ataupun 1. Citra biner didapatkan melalui konversi citra *grayscale* yang dinamakan proses *thresholding*. Dalam proses ini, akan diambil sebuah nilai *threshold* yang akan dijadikan sebagai acuan untuk proses konversi. Jika nilai intensitas sebuah piksel sama dengan atau lebih besar dari nilai *threshold*, maka nilai intensitas piksel tersebut akan dikonversi menjadi 1. Dan sebaliknya jika nilai intensitas piksel kurang dari nilai *threshold*, maka nilai intensitas piksel tersebut akan dikonversi menjadi 0. Sebagai contoh, 128 akan ditetapkan sebagai nilai *threshold*, jika piksel memiliki nilai intensitas sama dengan atau lebih dari 128, maka akan dikonversikan menjadi 1 atau

putih. Dan sebaliknya jika piksel tersebut memiliki nilai intensitas kurang dari 128, maka akan dikonversikan menjadi 0 atau hitam. Pada Gambar 2.4 akan ditunjukkan Gambar 2.3 yang telah dikonversikan menjadi citra biner.



Gambar 2.4 Citra Hasil Konversi RGB Menjadi Biner [6]

2.2 Digital Image Processing

Pengolahan citra digital (*digital image processing*) mempelajari cara sebuah citra dibentuk, dianalisis, dan diolah sedemikian rupa agar menghasilkan data-data atau informasi yang dapat dipahami secara manusiawi. Salah satu kegunaan pengolahan citra yaitu untuk memperbaiki kesalahan sinyal pada gambar atau piksel yang rusak yang disebabkan oleh adanya gangguan pada saat pembuatan gambar tersebut. Pengolahan citra digunakan untuk mempermudah manusia untuk melihat suatu gambar yang kabur agar terlihat lebih manusiawi dan gampang dilihat. Pengolahan citra dikelompokkan menjadi 3 bagian yaitu: *Image Enhancement*, *Image Restoration*, dan *Image Compression* [7].

Image Enhancement atau peningkatan kualitas gambar merupakan tahap dimana kontras pada gambar diperbaiki agar tingkat keabuaannya berada pada titik

yang normal sehingga lebih gampang untuk diinterpretasikan manusia. Dalam proses *image enhancement* terdapat beberapa teknik yaitu [7]:

1. Operasi titik

Operasi titik dalam peningkatan kualitas gambar dilakukan dengan cara menggantikan data-data dari citra dengan data-data yang baru secara manual agar sesuai dengan hasil yang diinginkan.

2. Operasi spasial

Operasi spasial dalam peningkatan kualitas gambar dilakukan dengan cara menggantikan nilai sebuah piksel yang akan diganti berdasarkan nilai-nilai median piksel lain yang disekitarnya sehingga diperoleh sebuah nilai piksel baru.

3. Operasi transformasi

Operasi transformasi dalam peningkatan kualitas gambar digunakan dengan cara memindahkan suatu citra asal ke tempat yang baru sesuai dengan keinginan.

Pada saat ini, teknologi pengolahan citra digital banyak digunakan dimanfaatkan dalam AI ataupun dalam kehidupan sehari-hari. Contoh beberapa aplikasi pengolahan citra digital yaitu [7]:

1. Aplikasi deteksi, klasifikasi dan identifikasi objek.

Aplikasi ini digunakan untuk dalam pendeteksi, pengklasifikasi dan pengidentifikasi objek, seperti aplikasi yang dipakai untuk mendeteksi objek seperti mobil atau sepeda.

2. Aplikasi pengenalan pola.

Aplikasi ini merupakan aplikasi yang digunakan digunakan untuk mengklasifikasi objek berdasarkan ciri-ciri yang dimiliki citra ke dalam beberapa kategori atau kelas.

3. *Machine Vision*

Aplikasi ini mengembangkan secara lanjut aplikasi pengenalan pola. Mesin ini menggunakan kamera untuk mengambil gambar dari sebuah objek dengan tujuan untuk menganalisa dan mendeskripsikan objek yang diambil kamera.

4. *Character Recognition (OCR)*

OCR juga mengembangkan aplikasi pengenalan pola. OCR ini biasanya digunakan dalam bidang informasi dan otomatisasi. Sistem OCR ini dibangun dengan alat-alat yang berubah lensa *sscanner*, pembangkit cahaya dan detektor.

5. *Computer Aided Diagnosis*

Computer Aided Diagnosis biasanya digunakan untuk mendiagnosa atau mengambil keputusan untuk membantu dokter

6. *Speech Recognition*

Speech Recognition atau pengenalan pola suara manusia digunakan agar komputer dapat mengenali bahasa manusia. Aplikasi ini biasa diterapkan dalam industri untuk meningkatkan efisiensi.

7. *Face Recognition*

Face Recognition atau pengenalan wajah merupakan aplikasi yang digunakan untuk mengenali wajah manusia. Aplikasi ini biasanya digunakan dalam bidang keamanan.

8. *Hand Gesture Recognition*

Hand gesture recognition merupakan teknologi yang mampu membaca gerak tangan kemudian dirubah menjadi teks dan atau suara. *Gesture recognition* bertujuan agar komputer bisa memahami gerakan manusia yang umumnya berasal dari tangan atau wajah

9. *Biometrics*

Biometrics dipakai untuk mengidentifikasi sebuah karakteristik unik yang dimiliki setiap individual.

10. *Image Database Retrieval*

Image Database Retrieval merupakan sistem komputer untuk menelusuri, mencari, dan mengambil gambar dari *database* besar gambar digital. Metode yang paling umum digunakan adalah menambahkan metadata ke teks, kata kunci, dan sebagainya.

11. *Data Mining*

Data mining adalah proses yang digunakan oleh perusahaan untuk mengubah data mentah menjadi informasi yang berguna. Dengan menggunakan perangkat lunak untuk mencari pola dalam kumpulan data yang besar, bisnis dapat mempelajari lebih lanjut tentang pelanggan mereka untuk mengembangkan strategi pemasaran yang lebih efektif, meningkatkan penjualan, dan menurunkan biaya.

2.3 *Computer Vision*

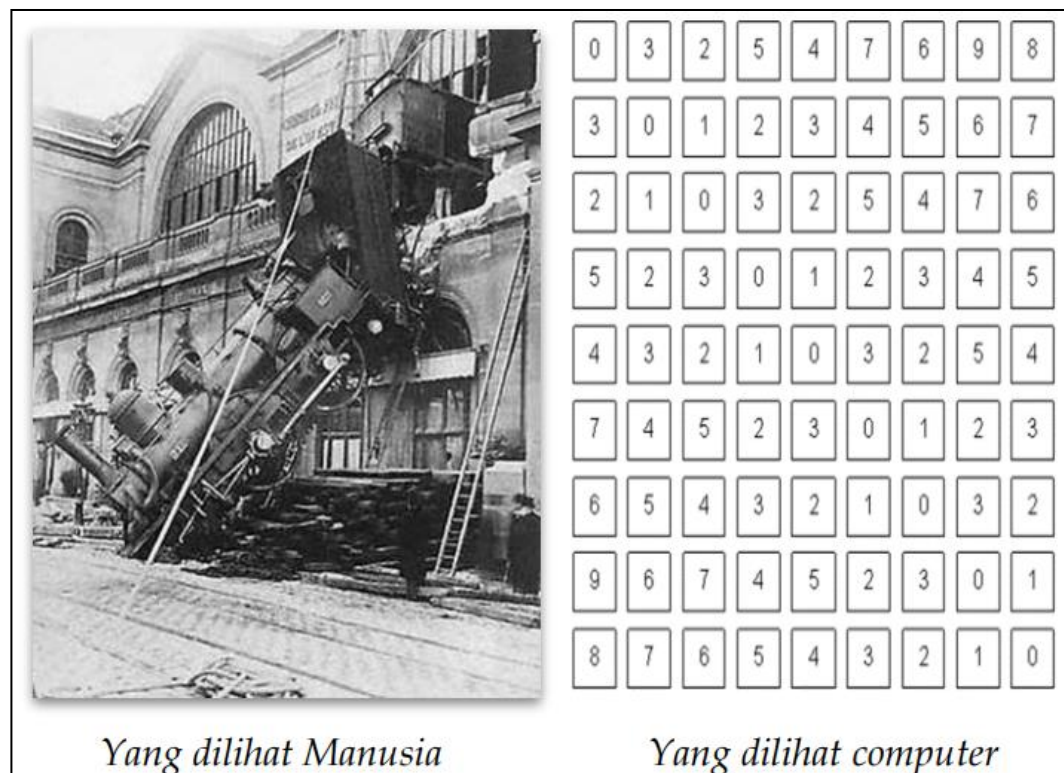
Computer vision atau visi komputer adalah bidang ilmu komputer yang berfokus pada replikasi bagian dari kompleksitas sistem penglihatan manusia dan

memungkinkan komputer untuk mengidentifikasi dan memproses objek dalam gambar dan video dengan cara yang sama seperti yang dilakukan manusia. Dari perspektif teknik, *computer vision* dapat digunakan untuk melakukan tugas-tugas yang dilakukan oleh penglihatan manusia secara otomatis. Tugas *computer vision* yaitu meliputi metode untuk memperoleh, memproses, menganalisis dan memahami gambar digital, serta ekstraksi data dimensi tinggi dari dunia nyata untuk menghasilkan informasi numerik atau simbolis, misalnya, dalam bentuk keputusan [7].

Sensor ini melampaui mata manusia karena mereka dapat melihat hal-hal yang tidak terlihat oleh manusia tetapi terbatas dalam aspek lain. Komputer dapat diprogram untuk memahami cukup banyak dari gambar yang diambil, seringkali hal-hal yang tidak dilihat atau dipedulikan manusia. Namun, komputer masih jauh dari kemampuan otak manusia. Kesenjangan menjadi semakin kecil seiring kemajuan teknologi dan sains. Aplikasi *computer vision* sangat luas dan mencakup semua aspek kegiatan sosial dan ekonomi: pengenalan pola, penilaian kualitas tak rusak untuk berbagai jenis produk, deteksi objek 3D, pertanian, cuaca, perawatan kesehatan, identifikasi, pengawasan, keselamatan publik, pembantu mengemudi, kendaraan *self-driving*, dll. Komputer sudah dapat memperkirakan cacat dan sifat produk/objek seperti warna, bentuk, ukuran, permukaan, kontaminasi, dan mereka dapat mengidentifikasi orang dan melacaknya. Batas ke mana visi komputer dapat pergi tidak dapat ditentukan, tetapi perkembangan saat ini menunjukkan batas hanya diberikan oleh imajinasi manusia dan segera komputer bahkan mungkin melihat lebih baik daripada mata manusia [7].

Transformasi gambar secara visual biasanya membutuhkan input yang memiliki ciri-ciri gambar agar dapat berinteraksi dengan proses pemikiran sistem agar tindakan yang sesuai dapat dieksekusi. Transformasi gambar ini melibatkan data gambar yang diperoleh melalui model yang dibangun dengan menggunakan campuran dari ilmu-ilmu lain, contohnya yaitu: geometri, statistika, dan juga algoritma komputer [7].

Computer vision ini terbagi menjadi beberapa sub bagian yaitu deteksi peristiwa, perkiraan gambar 3D, perkiraan gerakan, pengenalan objek dan juga pemulihan gambar. Gambar dibawah ini merupakan ilustrasi cara komputer melihat sebuah citra [7].



Gambar 2.5 Ilustrasi Komputer Melihat Citra [7]

2.4 Kecerdasan Buatan

Kecerdasan buatan atau yang sering dikenal dengan *Artificial Intelligence* (AI) merupakan kemampuan mesin atau sistem komputer untuk meniru atau menunjukkan kecerdasan manusia. Definisi ini melibatkan kemampuan sistem untuk mengumpulkan informasi, memahami konteks, melakukan analisis, membuat keputusan, dan belajar dari pengalaman untuk menghadapi tugas-tugas yang kompleks. Tujuan utama AI adalah untuk mengembangkan sistem yang dapat melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Beberapa tujuan utama AI meliputi [8]:

1. Pengolahan Bahasa Alami (*Natural Language Processing*).

AI bertujuan untuk mengembangkan sistem yang mampu memahami, menghasilkan, dan berinteraksi dengan bahasa manusia secara alami. Hal ini melibatkan kemampuan sistem untuk memahami konteks, sintaksis, semantik, dan pragmatik dalam bahasa manusia.

2. Pembelajaran Mesin (*Machine Learning*).

AI bertujuan untuk mengembangkan algoritma dan teknik yang memungkinkan mesin untuk belajar dari data dan pengalaman, tanpa diprogram secara eksplisit. Pembelajaran mesin melibatkan identifikasi pola, membuat prediksi, dan mengoptimalkan kinerja sistem berdasarkan data yang diberikan.

3. Pengenalan Pola (*Pattern Recognition*).

AI bertujuan untuk mengembangkan sistem yang dapat mengenali pola dan fitur dalam data, seperti pengenalan wajah, pengenalan suara, dan pengenalan

tulisan tangan. Ini melibatkan penggunaan algoritma dan teknik untuk mengidentifikasi dan mengklasifikasikan pola-pola yang ada dalam data.

4. Penalaran dan Pengambilan Keputusan (*Reasoning and Decision Making*).

AI bertujuan untuk mengembangkan sistem yang dapat menganalisis informasi, melakukan penalaran, dan membuat keputusan berdasarkan pemahaman konteks dan aturan yang ditentukan. Ini melibatkan penggunaan logika formal, inferensi, dan pemodelan pengetahuan untuk memungkinkan sistem AI untuk mengambil keputusan yang masuk akal.

Komponen utama dalam sistem AI meliputi [8]:

1. Basis Pengetahuan (*Knowledge Base*).

Tempat penyimpanan informasi dan pengetahuan yang digunakan oleh sistem AI. Basis pengetahuan dapat berisi fakta, aturan, konsep, dan model yang relevan dengan domain atau tugas yang sedang dihadapi.

2. Mesin Inferensi (*Inference Engine*).

Mesin inferensi bertanggung jawab untuk melakukan proses logika dan inferensi berdasarkan informasi yang ada dalam basis pengetahuan. Ini melibatkan penerapan aturan, penalaran, dan manipulasi pengetahuan untuk mencapai kesimpulan atau membuat keputusan.

3. Antarmuka Pengguna (*User Interface*).

Antarmuka pengguna memungkinkan interaksi antara pengguna manusia dan sistem AI. Ini dapat berupa antarmuka grafis, perintah suara, atau interaksi berbasis teks yang memungkinkan pengguna untuk memberikan input, menerima output, dan berkomunikasi dengan sistem AI.

4. Algoritma dan Teknik Kecerdasan Buatan (*AI Algorithms and Techniques*).

Ada berbagai algoritma dan teknik yang digunakan dalam AI, termasuk pembelajaran mesin, jaringan saraf tiruan, logika proposisional, pemrosesan bahasa alami, dan banyak lagi. Algoritma ini memungkinkan sistem AI untuk memproses data, mengenali pola, belajar dari pengalaman, dan membuat keputusan.

Ada beberapa jenis AI yang digunakan dalam berbagai konteks dan aplikasi. Dalam hal ini, kita akan menjelajahi beberapa jenis utama AI, termasuk sistem berbasis aturan, sistem pakar, pembelajaran mesin, jaringan saraf tiruan, dan AI yang terkait dengan robotika. Berikut adalah beberapa jenis AI yang umum [8]:

1. Sistem Berbasis Aturan.

Jenis AI ini menggunakan aturan-aturan yang ditentukan sebelumnya untuk menghasilkan output berdasarkan input yang diberikan. Sistem berbasis aturan terdiri dari himpunan aturan dan basis pengetahuan yang menggambarkan hubungan antara *input* dan *output*. Sistem ini bekerja dengan mencocokkan input dengan aturan yang sesuai untuk menghasilkan *output* yang diinginkan.

2. Sistem Pakar.

Sistem pakar adalah jenis AI yang menggabungkan pengetahuan ahli manusia dengan kemampuan komputasi untuk memecahkan masalah yang kompleks dalam domain tertentu. Sistem ini beroperasi dengan mengumpulkan pengetahuan dari para ahli manusia dan memanfaatkannya dalam membuat keputusan atau memberikan nasihat yang sesuai. Sistem pakar digunakan dalam bidang-bidang seperti kedokteran, keuangan, dan teknik.

3. Pembelajaran Mesin.

Pembelajaran mesin adalah cabang AI yang melibatkan pengembangan algoritma dan model statistik yang memungkinkan sistem untuk belajar dari data dan pengalaman. Dalam pembelajaran mesin, sistem menganalisis data untuk mengidentifikasi pola dan membuat prediksi atau keputusan berdasarkan informasi yang diberikan. Algoritma pembelajaran mesin dapat diklasifikasikan menjadi beberapa kategori, termasuk pembelajaran terawasi, pembelajaran tak terawasi, dan pembelajaran penguatan.

4. Jaringan Saraf Tiruan.

Jaringan saraf tiruan (*artificial neural networks*) adalah jenis AI yang terinspirasi oleh struktur dan fungsi jaringan saraf dalam otak manusia. Jaringan saraf tiruan terdiri dari banyak unit pemrosesan sederhana yang disebut neuron buatan, yang saling terhubung dan berinteraksi untuk memproses informasi. Jaringan saraf tiruan dapat digunakan untuk tugas-tugas seperti pengenalan pola, pengenalan wajah, dan pemrosesan bahasa alami.

5. AI dalam Robotika.

AI juga digunakan dalam pengembangan robotika, di mana sistem AI dikombinasikan dengan fisik robot untuk menghasilkan perilaku yang cerdas dan adaptif. Robotika AI melibatkan penggunaan algoritma pengolahan sensorik, perencanaan gerakan, dan pembelajaran untuk memungkinkan robot berinteraksi dengan lingkungan sekitarnya dan melakukan tugas-tugas tertentu.

2.5 *Machine Learning*

Machine Learning (ML) adalah cabang ilmu komputer yang berkembang dari bidang pengenalan pola dan teori pembelajaran komputer dalam kecerdasan

buatan. ML mempelajari konstruksi dan analisis algoritma pembelajaran dan prediksi data. Algoritma ML membangun model berdasarkan *input* model untuk mendapatkan sebuah prediksi atau keputusan yang didasarkan dari data daripada mengikuti instruksi program statis secara ketat. ML dikategorikan dalam 3 kategori besar yaitu [9]:

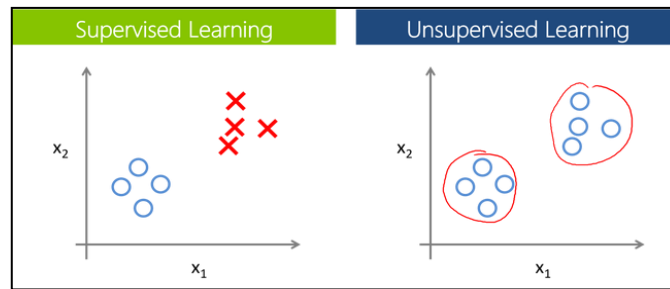
1. *Supervised Learning.*

Model atau algoritma disajikan dengan contoh masukan dan keluaran yang diinginkan kemudian dicari pola dan hubungan antara masukan dan keluaran tersebut. Tujuannya adalah untuk mempelajari aturan umum yang memetakan *input* ke *output*. Proses pelatihan berlanjut hingga model mencapai tingkat akurasi yang diinginkan pada data pelatihan. Beberapa contoh implementasi dari *Supervised Learning* di kehidupan nyata adalah:

- a. Klasifikasi gambar dimana dilakukan pelatihan dengan menggunakan gambar beserta label yang sesuai. Tujuannya adalah agar komputer dapat mengenali objek pada gambar yang baru di masa yang akan datang.
- b. Prediksi dimana komputer dilatih menggunakan data harga saham sebelumnya untuk menghasilkan prediksi harga di masa depan.

2. *Unsupervised Learning.*

Pada *Unsupervised Learning* data tidak diberikan label pada algoritma pembelajaran, dimana algoritma tersebut berusaha menemukan pola sebagai masukannya. Implementasi *Unsupervised Learning* dapat dimanfaatkan dalam mengelompokkan populasi yang memiliki perbedaan. *Unsupervised Learning* dapat mengelompokkan data tersendiri dan menemukan pola tersembunyi yang berada di dalam data.



Gambar 2.6 Perbedaan *Supervised Learning* dan *Unsupervised Learning* [10]

Seperti yang dapat dilihat pada Gambar 2.6, data dalam *Supervised Learning* diberi label, sedangkan data dalam *Unsupervised Learning* tidak diberi label.

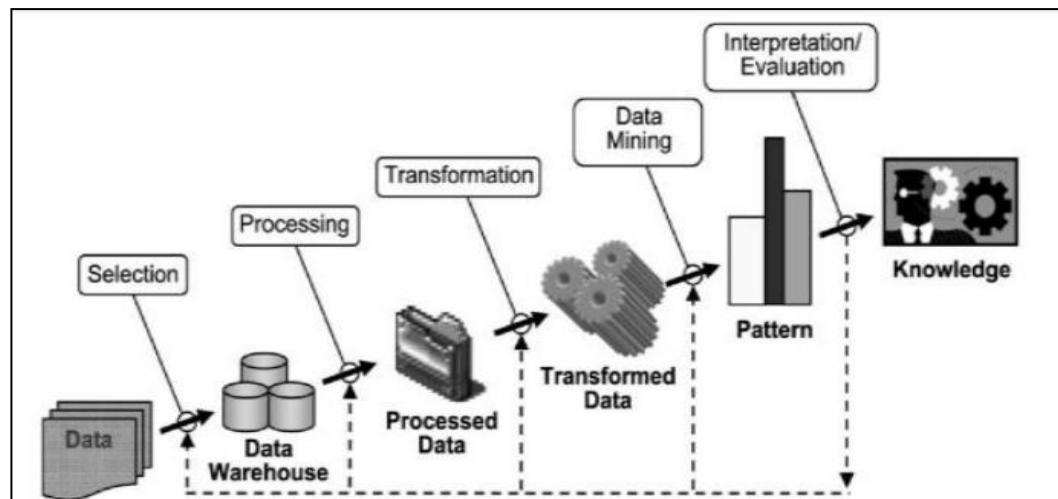
3. *Reinforcement Learning*.

Suatu bidang interdisipliner dalam pembelajaran mesin dan kontrol optimal yang berkaitan dengan bagaimana suatu agen cerdas dapat mengambil aksi di lingkungan yang dinamis dalam rangka untuk memaksimalkan penghargaan kumulatif. Dalam *Reinforcement Learning*, pengembang merancang metode untuk menghargai perilaku yang diinginkan dan menghukum perilaku negatif. Metode ini memberikan nilai positif pada tindakan yang diinginkan untuk mendorong agen dan nilai negatif pada perilaku yang tidak diinginkan. Metode ini memprogram agen untuk mencari imbalan keseluruhan jangka panjang dan maksimum untuk mencapai solusi yang optimal.

2.6 *Knowledge Discovery*

Knowledge Discovery atau *Knowledge Discovery in Database* (KDD) adalah salah satu metode untuk memperoleh pengetahuan dari basis data yang memiliki tabel-tabel yang saling berhubungan atau berelasi. Hasil dari pengetahuan yang didapatkan dari proses tersebut menjadi dasar sebagai basis pengetahuan

(*Knowledge Base*) dalam pengambilan keputusan. Berikut ini Gambar 2.7 menunjukkan proses *Knowledge Discovery* pada *database* [10].



Gambar 2.7 Proses *Knowledge Discovery* Pada *Database* [9]

Secara detail setiap tahapan proses pada *Knowledge Discovery* dapat dijelaskan sebagai berikut [10]:

1. *Data Selection.*

Tahapan pertama untuk menggali informasi menggunakan KDD yaitu dengan memilih atau menyeleksi beberapa data yang dibutuhkan dari sekumpulan data yang banyak. Data yang telah dipilih akan digunakan pada tahapan data mining. Untuk memudahkan penggunaan dan pencarian kembali data yang sudah diseleksi, data tersebut akan disimpan pada suatu berkas yang dibedakan dengan data lainnya.

2. *Pre-processing (Cleaning).*

Untuk memastikan data yang disimpan merupakan data yang bermutu dan akurat, dilakukan tahapan ini yaitu dengan membuang data yang tidak lengkap atau tidak valid. Tahapan ini penting agar proses data mining nanti dapat

menghasilkan data yang bermutu dan valid sesuai dengan data yang dimasukkan.

3. *Transformation.*

Pada tahapan transformasi data, data akan disesuaikan dengan teknik data mining yang akan dipakai. Diterapkan suatu format pada data sehingga nanti data yang tidak sesuai dengan format akan disingkirkan dan menyisakan data yang sesuai untuk digunakan pada tahapan selanjutnya dengan kualitas data yang tepat.

4. *Data Mining.*

Pada proses data *mining*, digunakan beberapa teknik atau metode tertentu untuk menganalisis data yang ada sehingga didapatkan hasilnya yaitu data yang berisi pengetahuan penting atau tersembunyi. Pemilihan teknik harus sesuai dengan tujuan KDD dilakukan sehingga hasilnya akan relevan.

5. *Evaluasi Pola (Pattern Evaluation).*

Hasil data yang didapatkan dari proses data mining akan diidentifikasi sehingga didapatkan suatu pola yang akan dimasukkan ke dalam *knowledge based* untuk kemudian dilakukan analisis.

6. *Presentasi Pengetahuan (Knowledge Presentation).*

Pada tahapan ini, data yang ada di dalam *knowledge based* tersebut akan ditampilkan atau diperlihatkan terkait metode yang digunakan untuk memperolehnya kepada pengguna yang melakukan *Knowledge Discovery* tersebut.

2.7 Metode *Hand Gesture Recognition*

Hand Gesture Recognition atau gerakan yang diketahui/dikenal adalah sebuah teknologi yang membuat sebuah gerakan dari objek yang telah diidentifikasi contohnya seperti tangan, kaki, kepala, tubuh ataupun lain-lain menjadi sebuah perintah yang akan dikenali oleh mesin, aplikasi, komputer karena telah diotomisasi. Disaat gerakan atau objek tersebut telah dikenali maka perintah akan dijalankan. *Hand Gesture Recognition* adalah hasil dari proses otomasi yaitu diintegrasinya berbagai proses untuk visual yang diinginkan. Konsep kerja bagaimana sebuah objek atau citra dari sebuah gesture menjadi perintah adalah seperti mata manusia bekerja, mengolah, mengakuisisi, dan pengenalan pola dari citra atau objek tersebut menjadi sebuah keputusan [11].

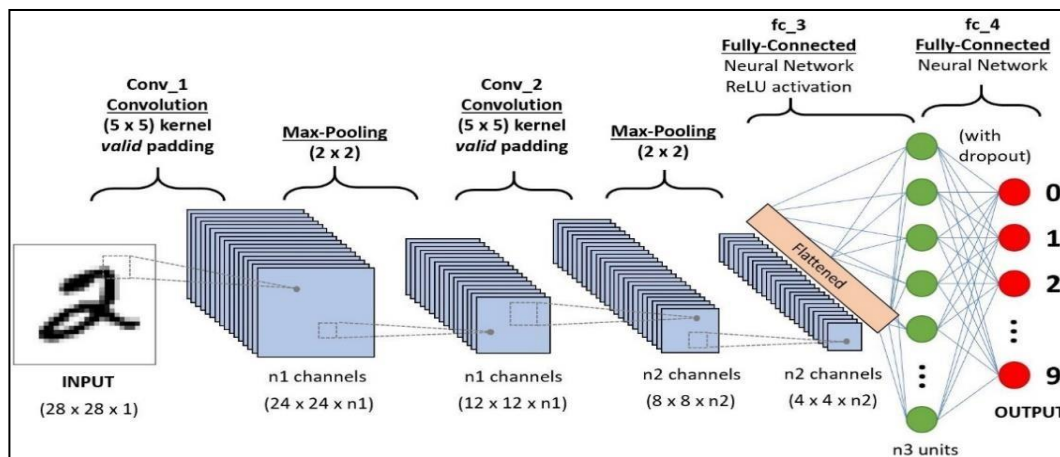
Gesture adalah bentuk komunikasi non-verbal dengan aksi tubuh yang mengkomunikasikan pesan-pesan tertentu, sebagai pengganti ucapan. Sistem pengenalan gesture tangan terus berkembang, terutama untuk menggantikan peran perangkat masukan seperti *mouse*, *keyboard* dan *joystick* yang digunakan pada interaksi manusia dengan komputer, namun perangkat ini memiliki kelemahan diantaranya tidak alami bagi manusia dan jarak pengontrolan masih cukup dekat. Sistem pengenalan *gesture* tangan mengimplementasikan cara kerja sistem penglihatan dan pengenalan yang melibatkan mata manusia sebagai sensor yang merekam objek berupa *gesture* tangan dan kemudian mengirim informasi ke otak untuk diproses sehingga menghasilkan sebuah *output* [11].

Pada penelitian ini metode *Hand Gesture Recognition* yang digunakan pada penelitian ini adalah dengan menggunakan *Convolutional Neural Network* (CNN).

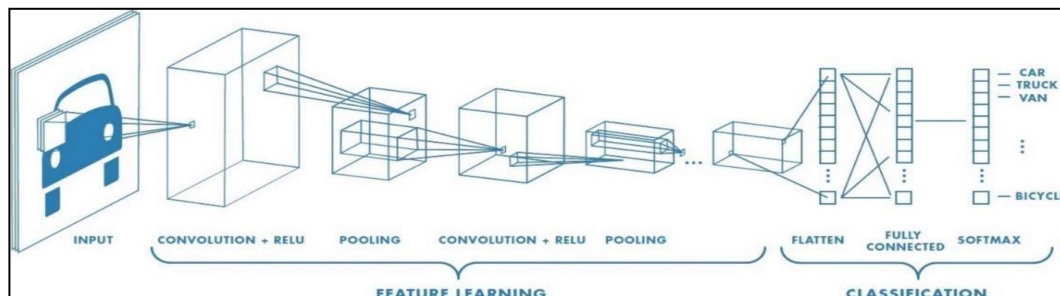
2.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah bagian dari *deep learning* yang membuat gambar untuk dikenali dan diklasifikasi. Metode CNN adalah sebuah kelas pada *neural network* yang berjalan dalam bidang pemrosesa data-data yang memiliki topologi dalam bentuk *grid*, contohnya yaitu gambar. Metode CNN dapat biasanya digunakan dalam analisa dokumen, klasifikasi gambar dan video, pengenalan wajah, dan lain lain [12].

Secara garis besar, CNN terdiri dari *neuron* yang memiliki *weight*, bias dan *activation function*. Arsitektur Convolutional Neural Network dapat dilihat pada Gambar 2.8 [13].



Gambar 2.8 Arsitektur Convolutional Neural Network [13]



Gambar 2.9 Ilustrasi Arsitektur Convolutional Neural Network [13]

Dari Gambar 2.9 dan Gambar 2.10, dapat dilihat bahwa arsitektur CNN

terbagi dalam dua bagian yaitu yang pertama *Feature Extraction layer* dan yang kedua yaitu *Fully Connected Layer*. Dalam *Feature Extraction Layer*, sebuah gambar akan dilakukan *encoding* untuk menjadi *feature map* yang memiliki nilai dan angka yang mewakili citra tersebut atau proses ini disebut juga *Feature Extraction*. *Feature Extraction Layer* ini kemudian dibagi menjadi dua yaitu *Convolution Layer* dan *Pooling Layer*. *Convolution Layer* merupakan bagian penting yang membedakan metode CNN dengan metode *neural network* lainnya dan *layer* ini juga merupakan bagian utama dari metode CNN. Pada *Convolution Layer*, citra yang dimasukkan akan diekstraksi ciri-cirinya sehingga dapat dikenali. Proses konvolusi (*Convolution*) menggunakan fungsi matematika antara matriks citra dengan *filter* atau *kernel* yang terdapat dalam *layer* ini. *Layer* ini mempelajari ciri citra dengan menjaga hubungan antara setiap *pixel*. *Kernel* merupakan operator yang dipakai untuk memperoleh sebuah *array* dari citra. Cara kerja *kernel* yaitu dengan diterapkan dalam seluruh citra. Ukuran matriks pada *kernel* biasanya berukuran 3*3 atau 5*5 yang berisi nilai acak antara -1 dan 1. Pada Gambar 2.10 ditunjukkan proses konvolusi pada sebuah gambar [13].

131	162	232	84	91	207
104	-1	109	+1	237	109
243	-2	202	+2	135	26
185	-15	200	+1	61	225
157	124	25	14	102	108
5	155	116	218	232	249

Gambar 2.10 Contoh Operasi Matematis Proses Konvolusi [13]

Hasil konvolusi matriks citra dengan *filter (kernel)* tersebut disebut *Feature Map*. Kemudian nilai negative pada *feature map* akan diganti menjadi nilai positif. Proses ini disebut ReLU atau *Rectified Linear Unit*, seperti yang ditampilkan pada Gambar 2.11 [13].

Input			ReLU		
-249	-91	-37	0	0	0
250	-134	101	250	0	101
27	61	-153	27	61	0

Gambar 2.11 Contoh Penerapan Aktivasi ReLU [13]

Dalam *Pooling Layer* atau *Max-Pooling*, ukuran spasial akan dikurangi agar dapat jumlah parameter dapat berkurang untuk ukuran citra-citra yang besar. *Pooling Layer* dijalankan pada setiap *feature map* secara independen. *Fully Connected Layer* adalah bagian dari *feature map* dari *Feature Extraction Layer* yang diubah menjadi vektor dalam bentuk *neural network* dari *array* multidimesi, proses pengubahan ini disebut juga *Flatten*. *Neural network* yang dibentuk kemudian akan dikombinasikan menjadi sebuah model. Akhirnya, dengan fungsi aktivasi *softmax* atau *sigmoid*, output dapat diklasifikasikan menjadi wajah manusia, ataupun hewan seperti kucing, anjing, dan sebagainya [13].

Secara singkat, dengan melalui serangkaian proses, CNN dapat melatih (*training*) dan menguji (*testing*) gambar agar dapat dikenali. Gambar yang dilatih akan diubah menjadi pecahan gambar yang lebih kecil. Kemudian gambar yang lebih kecil tersebut akan melalui *layer* yang akan menyimpan hasil dari gambar

yang sudah diperkecil dalam serangkaian *array* untuk mengurangi ukuran gambar dan jumlah parameter ketika perhitungan dilakukan. Kemudian akan dilakukan prediksi untuk mengklasifikasi gambar tersebut [13].

Metode CNN sering dipakai karena metode ini dapat mengekstraksi ciri-ciri penting dari sebuah gambar secara otomatis dan juga merupakan metode yang paling efisien dibandingkan metode *neural network* yang lain, utamanya dalam masalah kompleksitas dan juga memori. Namun metode ini juga memiliki kekurangan yaitu memerlukan banyak waktu melatih sebuah model, memerlukan banyak *data training* dan *data testing*, dan juga dapat terjadi *overfitting*. *Overfitting* terjadi ketika model tidak mampu melakukan generalisasi pada gambar karena banyaknya *noise* dalam gambar, model memiliki variasi yang tinggi, *data training* yang terlalu banyak ataupun model yang terlalu kompleks. [13].

2.9 Algoritma *Random Forest Classifier*

Random Forest Classifier merupakan sebuah metode *ensemble* yang terdiri atas sekumpulan *decision tree* (pohon keputusan) yang digunakan untuk klasifikasi data ke suatu kelas. Langkah awal untuk menentukan pohon keputusan adalah dengan menghitung nilai *entropy* dan *information gain*. Terdapat dua buah formula dari algoritma *Random Forest Classifier* dalam membentuk pohon keputusan antara lain [14]:

1. Entropi.

Pendekatan entropi sebagai penentu ketidakmurnian atribut. Penghitungan nilai entropi dapat dilihat pada persamaan berikut:

$$Entropy(S) = \sum_{i=1}^n -p_i \log_2 p_i \quad (2.1)$$

Keterangan:

S : Himpunan dataset

n : Banyaknya jumlah kelas

p_i : Proporsi dari S_i terhadap S

2. *Information Gain.*

Information gain merupakan nilai perolehan informasi dalam memilah simpul.

Perhitungan *gain* sangat berpengaruh terhadap setiap node teratas dan node pemisah. Perhitungan gini masih berlanjut ketika hasil akhir gini masih berupa angka dan berhenti ketika hasil akhir gini adalah nol.

$$\text{Information Gain}(A) = \text{Entropy}(S) = \sum_{i=1}^n \frac{|S_i|}{|S|} * \text{Entropy}(S_i) \quad (2.2)$$

Keterangan:

A : Atribut

S : Himpunan dataset

$|S_i|$: Jumlah sampel untuk nilai ke-i

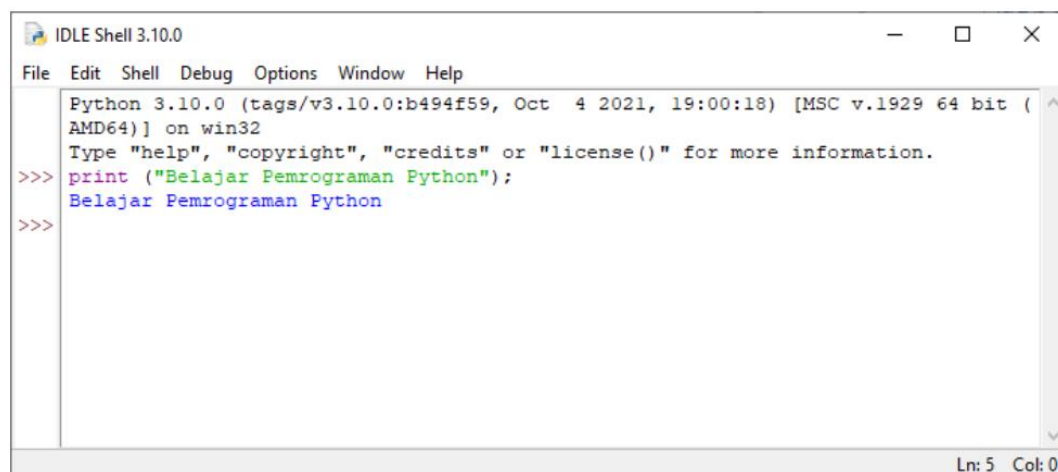
$|S|$: Banyaknya jumlah data

2.10 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python bisa dibilang bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat *source code* mudah dibaca.

Python juga memiliki *library* yang lengkap sehingga memungkinkan programmer untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana [7].

Untuk menjalankan bahasa pemrograman Python dapat menggunakan *Integrated Development and Learning Environment* (IDLE) Python. Aplikasi IDLE merupakan halaman untuk pengembangan Python, terdiri dari editor dan interpreter bahasa pemrograman Python. Berikut ini, Gambar 2.12 menunjukkan IDLE Python [7].



Gambar 2.12 IDLE Python [7]

2.11 Penelitian Terdahulu

Penelitian yang membahas mengenai implementasi teknologi informasi dalam penerjemahan bahasa isyarat sudah pernah dilakukan sebelumnya yaitu pada tahun 2022 yang membahas mengenai perancangan aplikasi penerjemah bahasa isyarat ke dalam huruf abjad menggunakan algoritma klasifikasi *Convolutional Neural Network* (CNN). Hasil penelitian tersebut menyimpulkan bahwa algoritma CNN dalam menerjemahkan bahasa isyarat mendapatkan akurasi 0,995 atau 99,5%

[4]. Penelitian terbaru yaitu di tahun 2023 mengimplementasikan algoritma *Long-Short Term Memory* (LSTM) dalam pengembangan aplikasi penerjemah bahasa isyarat. Hasil penelitian menunjukkan bahwa algoritma LSTM mendapatkan akurasi di antara 75%-85% [3].

BAB III

METODE PENELITIAN

3.1 Analisis Masalah

Dalam subbab ini, akan dilakukan analisis secara mendalam terhadap masalah yang sedang dihadapi. Langkah ini penting untuk memahami akar permasalahan secara menyeluruh sebelum kita mencari solusi yang tepat. Dengan mengurai masalah menjadi komponen-komponen utamanya, dapat diidentifikasi faktor-faktor yang berkontribusi pada masalah tersebut. Melalui analisis ini, diharapkan kita dapat memperoleh wawasan yang lebih jelas tentang sifat dan ruang lingkup masalah yang dihadapi, serta membuka jalan bagi strategi penyelesaian yang efektif dan berkelanjutan. Analisis masalah pada penelitian ini dikategorikan menjadi 3 yaitu analisis proses yang berjalan sesuai di lapangan, analisis metode yang dan algoritma yang digunakan, dan analisis sistem usulan.

3.1.1 Analisis Proses Yang Berjalan Sesuai di Lapangan

Analisis proses yang berjalan menguraikan secara mendalam tahapan sistem berjalan yang digunakan saat ini dalam proses menerjemahkan bahasa isyarat. Berikut ini uraian proses berjalan sesuai di lapangan yang digunakan antara lain:

1. Pada praktiknya dalam menerjemahkan bahasa isyarat biasanya dibantu oleh seorang penerjemah. Namun penggunaan penerjemah terkadang kurang efisien dan memerlukan biaya besar. Kebutuhan akan penerjemah bahasa isyarat menambah lapisan tambahan dalam komunikasi, yang dapat memperlambat

aliran informasi dan menghambat interaksi langsung antara penutur bahasa isyarat dan penerima pesan. Proses penerjemahan sendiri memerlukan waktu tambahan, karena penerjemah harus mendengarkan pesan, memahaminya, dan kemudian menerjemahkan ke dalam bahasa isyarat, yang dapat mengakibatkan keterlambatan dalam respons. Selain itu memerlukan biaya yang besar untuk menyewa seorang penerjemah bahasa isyarat.

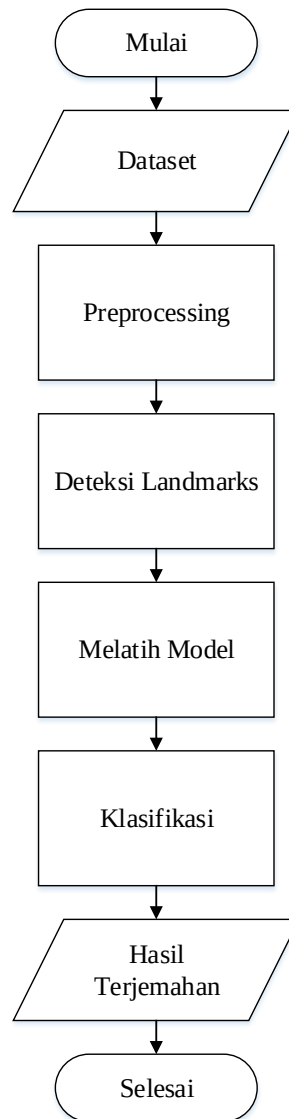
2. Penggunaan teknologi informasi dalam membantu proses penerjemahan bahasa isyarat. Namun menggunakan penggunaan teknologi informasi masih memiliki kekurangan yakni kesulitan dalam pengenalan isyarat, keterlambatan dalam respons, kesulitan menangkap konteks dan nuansa, ketergantungan pada koneksi internet, serta masalah privasi dan keamanan.

Berdasarkan hasil analisis proses yang berjalan sesuai di lapangan ditemukan beberapa permasalahan yang harus dicari solusinya. Solusi dari penelitian ini yaitu dengan membangun sebuah aplikasi penerjemah bahasa isyarat menggunakan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier*.

3.1.2 Analisis Metode dan Algoritma Yang Digunakan

Analisis metode dan algoritma yang digunakan berisikan tahapan-tahapan yang berupa pengauran cara kerja dari metode *Hand Gesture Recognition* dalam mendeteksi *gesture* jari dan cara kerja serta perhitungan manual dari algoritma *Random Forest Classifier* dalam melakukan klasifikasi terhadap data-data yang ada sebelumnya untuk menerjemahkan bahasa isyarat. Berikut ini, Gambar 3.1

menunjukkan *flowchart* alir penelitian dalam melakukan penerjemahan bahasa isyarat.



Gambar 3.1 *Flowchart* Alir Penelitian

Berdasarkan Gambar 3.1 diuraikan tahapan penelitian dari cara kerja metode dan algoritma yang digunakan antara lain:

1. *Dataset*.

Tahap di mana data bahasa isyarat dikumpulkan. *Dataset* ini akan menjadi dasar untuk melatih model pengenalan bahasa isyarat. *Dataset* pada penelitian

ini diambil dari dataset lokal dengan setiap kata bahasa isyarat masing-masing memiliki 100 gambar data latih. Contoh salah satu dataset yang digunakan dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 Contoh *Dataset* Penelitian

2. *Preprocessing*.

Tahap *preprocessing* pada penelitian ini dilakukan dengan cara melakukan proses *resize* terhadap *dataset* gambar menjadi ukuran 200x200 dan dilakukan proses konversi ke *grayscale* ketika proses *capture dataset* dilakukan. Asumsikan terdapat sebuah citra dengan ukuran 200x200 seperti Gambar 3.3 berikut.



Gambar 3.3 *Dataset* Bahasa Isyarat Makan

Kemudian dari citra RGB pada Gambar 3.2 akan diakses sebuah *pixel* secara *random* untuk mengetahui nilai RGB salah satu *pixel*. Pada *pixel* $x = 100$ dan $y = 100$ diketahui Nilai RGB = 57 34 16. Dari data tersebut maka dapat dihitung *Grayscale* dengan cara sebagai berikut :

Diketahui = *Red* (merah) = 57

Green (hijau) = 34

Blue (biru) = 16

Grayscale = $0.2989 R + 0.5870 G + 0.1140 B$

Grayscale = $(0.2989 \times 57) + (0.5870 \times 34) + (0.1140 \times 16)$

Grayscale = $17.0373 + 19.958 + 1.824$

Grayscale = 38.8193

Citra *Grayscale* merupakan citra dengan ukuran 8 bit. Sehingga memiliki ($2^8 - 1$) kemungkinan warna mulai dari 0 hingga 255 dimana 0 berarti hitam dan 255 berarti putih. Warna abu memiliki *range* dari 1 hingga 254 dimulai dari abu paling gelap hingga abu terang dan akhirnya mendekati putih. Jadi hasil *grayscale* = 38.8193 merupakan warna abu-abu yang cenderung mendekati hitam. Kemudian aplikasikan rumus tersebut di semua *pixel* dan hasil dari konversi citraRGB pada Gambar 3.3 akan menjadi seperti pada Gambar 3.4 berikut.



Gambar 3.4 Citra *Grayscale*

3. Deteksi *Landmarks*.

Selanjutnya, dataset yang telah melewati tahap *preprocessing* dilanjutkan pada tahapan berikutnya yaitu mendeteksi *landmarks* atau titik-titik penting pada

gerakan tangan dalam bahasa isyarat. Berikut ini uraian cara kerja deteksi *landmarks* yang digunakan:

a. Model *Hands*:

MediaPipe menyediakan model *Hands* yang digunakan untuk mendeteksi titik landmark tangan dalam citra.

b. Mendeteksi *Landmarks*:

Dengan menggunakan model *Hands*, citra masukan diproses untuk mendeteksi titik-titik *landmark* tangan.

Titik-titik *landmark* ini adalah titik-titik penting dalam tangan yang digunakan untuk mengidentifikasi pose tangan.

c. *Landmarks* pada tangan:

Setiap tangan dapat memiliki beberapa *landmarks*, yang mewakili titik-titik penting seperti ujung jari, pangkal jari, dan titik-titik lainnya yang membentuk pose tangan.

Jumlah total *landmarks* yang dideteksi tergantung pada pose tangan yang diidentifikasi dalam citra.

d. Penggunaan *Landmarks*:

Setelah mendeteksi *landmarks*, koordinatnya digunakan sebagai fitur masukan untuk model klasifikasi.

Dalam kasus ini, koordinat *landmarks* tangan digunakan sebagai fitur masukan untuk model klasifikasi bahasa isyarat.

Dengan menggunakan *landmarks* tangan sebagai fitur masukan, model dapat belajar untuk mengenali pola-pola yang berkaitan dengan bahasa isyarat dari

citra tangan yang diberikan. *Landmarks* ini menjadi representasi penting dari tangan dalam konteks pengenalan bahasa isyarat.

4. Melatih Model.

Setelah proses pelatihan *landmarks* dataset dilakukan, selanjutnya dilakukan pelatihan dengan menggunakan algoritma *Random Forest Classifier*. Asumsikan terdapat 3 kelas citra yaitu makan, tidur, dan minum. Berikut ini uraian cara kerja perhitungan manual dari algoritma *Random Forest Classifier* antara lain:

a. Memilih fitur terbaik:

Misalkan dipilih fitur pertama sebagai pemisah. Fitur ini adalah jumlah piksel hitam dalam gambar.

b. Memisahkan data:

Pertama, kita hitung jumlah piksel hitam dalam setiap gambar untuk semua kata.

Berdasarkan nilai fitur pertama (jumlah piksel hitam), dipisahkan dataset menjadi dua kelompok: satu dengan nilai fitur kurang dari atau sama dengan nilai ambang batas, dan satu lagi dengan nilai fitur lebih besar dari nilai ambang batas.

c. Membuat keputusan:

Jika jumlah piksel hitam kurang dari atau sama dengan nilai ambang batas, maka pergi ke simpul anak kiri.

Jika jumlah piksel hitam lebih besar dari nilai ambang batas, maka pergi ke simpul anak kanan.

d. Rekursi:

Langkah-langkah di atas diulangi untuk setiap simpul anak, baik yang kiri maupun kanan, hingga tidak ada lagi data atau kriteria pemberhentian lainnya terpenuhi.

Contoh perhitungan manual:

Misalkan memiliki *dataset* sebagai berikut (dalam jumlah piksel *grayscale*):

- a. "DONE": 100
- b. "HALO": 150
- c. "SORRY": 120

Misalkan dipilih nilai ambang batas (misalnya, 130).

- a. Karena $100 \leq 130$, kita tempatkan kata " DONE" di simpul anak kiri.
- b. Karena $150 > 130$, kita tempatkan kata "HALO" di simpul anak kanan.

Dengan demikian, didapatkan satu pohon keputusan sederhana yang memisahkan kata-kata tersebut berdasarkan jumlah piksel hitam dalam gambar.

5. Klasifikasi.

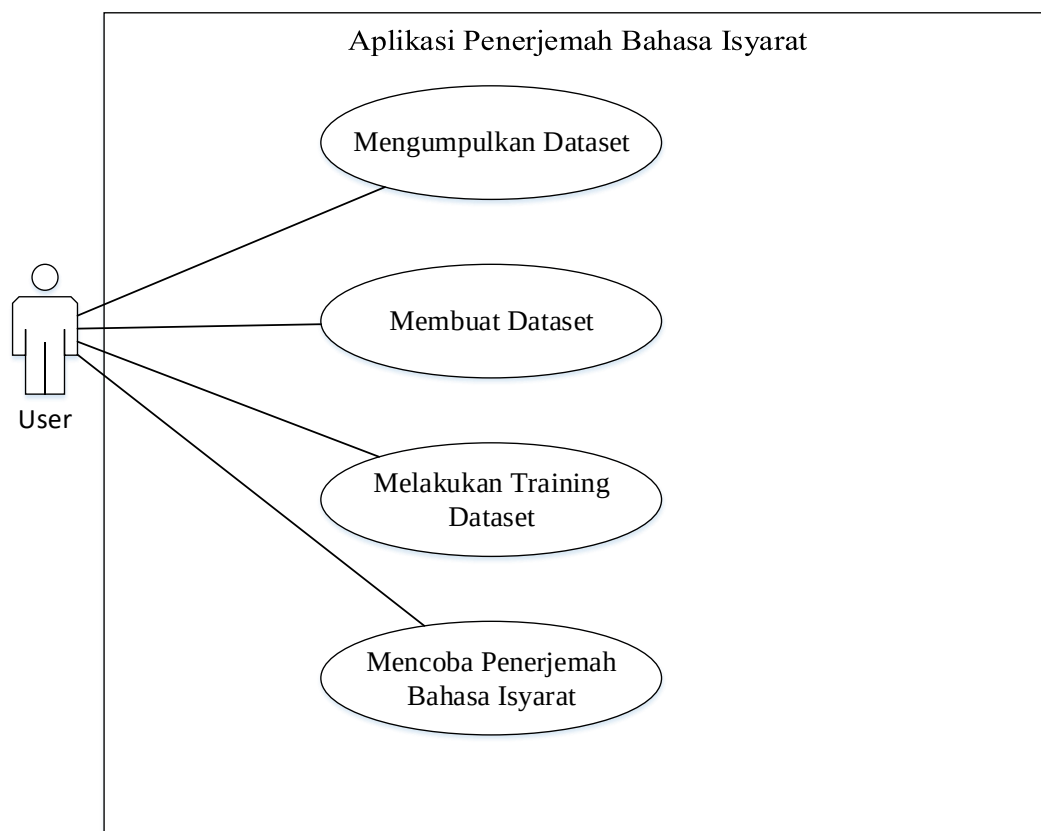
Dengan menggunakan *landmarks* yang terdeteksi dan model klasifikasi yang telah dibuat, maka selanjutnya diklasifikasikan bahasa isyarat ke dalam kategori yang sesuai. Ini adalah tahap di mana terjemahan dari bahasa isyarat dilakukan, di mana model mengidentifikasi isyarat yang dilakukan dan menentukan artinya berdasarkan kelas yang telah dipelajari selama pelatihan..

6. Hasil Terjemahan.

Output dari sistem, berupa hasil terjemahan dari bahasa isyarat. Hasil ini disajikan dalam bentuk representasi visual secara *real time* serta teks hasil terjemahan yang memungkinkan pengguna untuk memahami pesan yang disampaikan dalam bahasa isyarat.

3.1.3 Analisis Sistem Usulan

Dalam sub bab ini, akan dilakukan analisis mendalam terhadap sistem yang diusulkan dalam penelitian ini. Sistem usulan merupakan hasil dari pengembangan dan implementasi konsep yang telah dibahas sebelumnya. Analisis ini bertujuan untuk mengevaluasi kinerja, kehandalan, dan efektivitas sistem dalam memenuhi tujuan yang telah ditetapkan. Pemodelan sistem usulan menggunakan *Use Case Diagram*. Berikut ini, Gambar 3.5 menunjukkan *Use Case Diagram* sistem usulan.



Gambar 3.5 Use Case Diagram Aplikasi Penerjemah Bahasa Isyarat

Berdasarkan *Use Case Diagram* pada Gambar 3.1 tersebut diuraikan skenario dari setiap butir-butir *Use Case* yang diuraikan pada Tabel 3.1 hingga Tabel 3.4.

Tabel 3.1 Skenario *Use Case* Mengumpulkan Dataset

Mengumpulkan <i>Dataset</i>		Nama <i>Use Case</i>
<i>User</i>		Aktor
Melakukan pengumpulan <i>dataset</i> gerakan tangan bahasa isyarat		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman pengumpulan <i>dataset</i> dengan menekan menu <i>collect and create dataset</i>		Normal Flow Event
	2. Menampilkan halaman <i>collect and create dataset</i>	
3. Melakukan pengisian kelas <i>dataset</i> dan kemudian menekan tombol <i>collect dataset</i>		
	4. Menyalakan <i>webcam</i> dan mulai mengumpulkan <i>dataset</i> dengan cara menyimpan foto di depan objek <i>webcam</i> sebanyak 100 foto	

Tabel 3.2 Skenario *Use Case* Membuat Dataset

Membuat <i>Dataset</i>		Nama <i>Use Case</i>
<i>User</i>		Aktor
Melakukan pembuatan <i>landmarks dataset</i> dengan menggunakan metode <i>Hand Gesture Recognition</i>		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman pembuatan <i>dataset</i> dengan menekan menu <i>collect and create dataset</i>		Normal Flow Event

	2. Menampilkan halaman <i>collect and create dataset</i>	
3. Menekan tombol <i>create dataset</i>		
	4. Melakukan proses pembuatan <i>dataset</i> berupa pembuatan <i>landmarks</i> tangan dari <i>dataset</i> dengan menggunakan metode <i>Hand Gesture Recognition</i> yang disimpan ke <i>data.pickle</i> serta menampilkan pesan validasi proses pembuatan <i>dataset</i> berhasil	

Tabel 3.3 Skenario Use Case Melakukan Training Dataset

Melakukan Training Dataset		Nama Use Case
User		Aktor
Melakukan proses <i>training dataset</i> gerakan tangan bahasa isyarat untuk membuat model klasifikasi gerakan tangan bahasa isyarat		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman <i>training dataset</i> dengan menekan menu <i>training dataset</i>		Normal Flow Event
	2. Menampilkan halaman <i>training dataset</i>	
3. Menekan tombol <i>training</i>		
	4. Melakukan proses <i>training dataset</i> dengan menggunakan algoritma <i>Random Forest Classifier</i> yang disimpan ke <i>model.p</i>	

	serta menampilkan pesan validasi proses <i>training dataset</i> berhasil	
--	--	--

Tabel 3.4 Skenario Use Case Mencoba Penerjemah Bahasa Isyarat

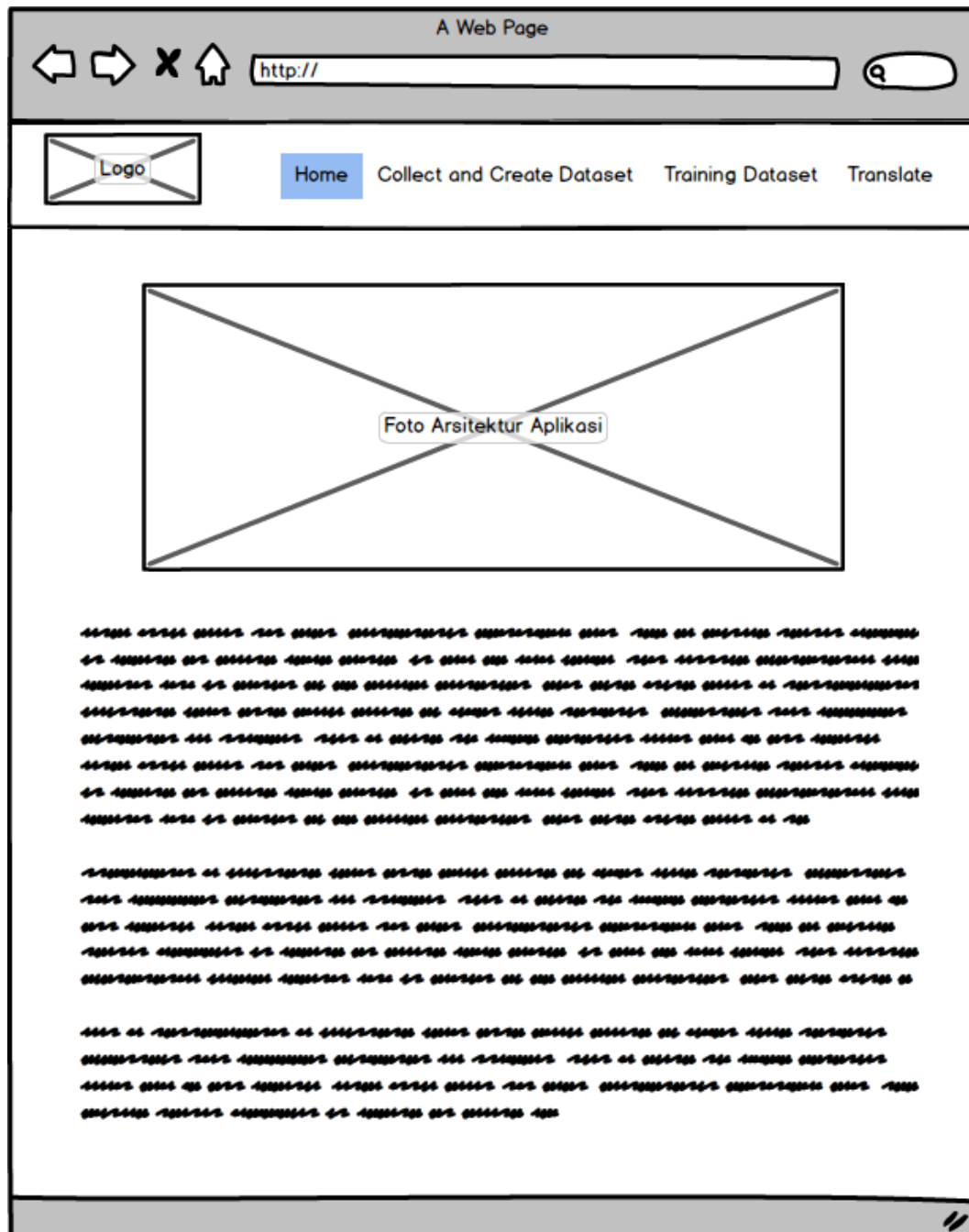
Mencoba Penerjemah Bahasa Isyarat		Nama Use Case
User		Aktor
Mencoba fitur penerjemah bahasa isyarat dengan kombinasi metode <i>Hand Gesture Recognition</i> dan algoritma <i>Random Forest Classifier</i>		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman fitur penerjemah bahasa isyarat dengan menekan menu <i>translate</i>		Normal Flow Event
	2. Menampilkan halaman penerjemah bahasa isyarat dengan <i>webcam</i> yang langsung menyala	
3. Melakukan gerakan isyarat di depan layar <i>webcam</i>		
	4. Mendeteksi <i>gesture</i> tangan dengan memetakannya dalam bentuk garis tangan serta menerjemahkan gerakan yang dibuat dalam bentuk teks jika sesuai dengan model klasifikasi	

3.2 Perancangan

Pada sub bab ini diuraikan seluruh rancangan tampilan dari aplikasi penerjemah bahasa isyarat yang akan dikembangkan pada penelitian ini. Rancangan aplikasi berupa aplikasi berbasis *website* dengan keseluruhan rancangannya sebagai berikut:

1. Rancangan Halaman Awal.

Halaman awal berisikan menu-menu navigasi aplikasi serta pengenalan mengenai model rancangan aplikasi serta penjelasannya sehingga memberikan pemahaman kepada pengguna aplikasi.



Gambar 3.6 Rancangan Halaman Awal

2. Rancangan Halaman *Collect and Create Dataset*.

Halaman *collect and create dataset* berisikan sebuah *form* untuk mengumpulkan *dataset* serta sebuah *button* untuk membuat *dataset* yaitu membuat *landmarks gesture* tangan dengan metode *Hand Gesture Recognition* terhadap *dataset* yang terkumpul.

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://". The navigation bar includes a "Logo" icon and four links: "Home", "Collect and Create Dataset" (highlighted in blue), "Training Dataset", and "Translate".

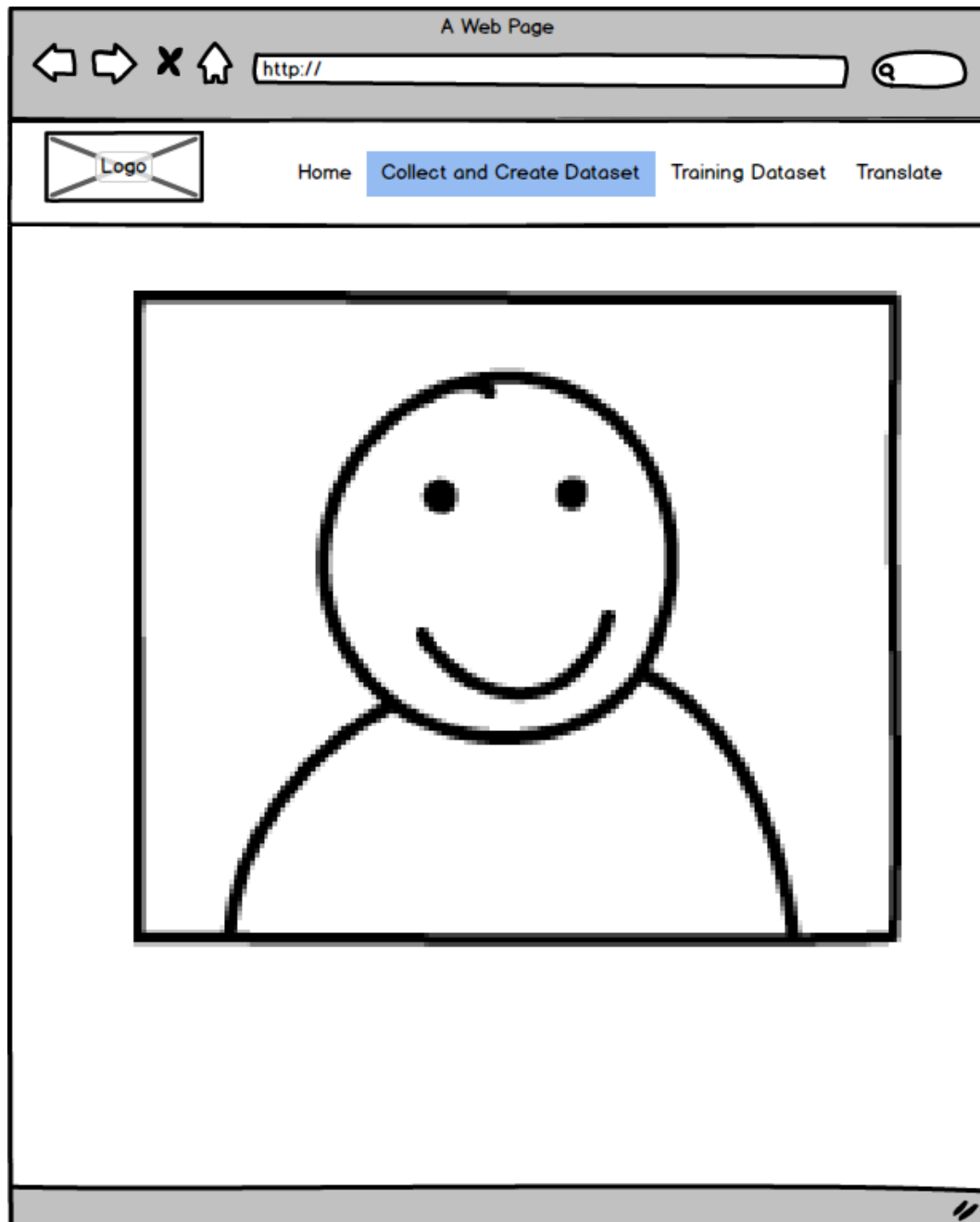
The main content area is divided into two sections:

- Collect Dataset**: This section contains a label "Class:" followed by a text input field and a "Collect" button.
- Create Dataset**: This section contains the text "Press this button if you want to create dataset" followed by a "Create Dataset" button.

Gambar 3.7 Rancangan Halaman *Collect and Create Dataset*

3. Rancangan Halaman *Collect Dataset Using Webcam*.

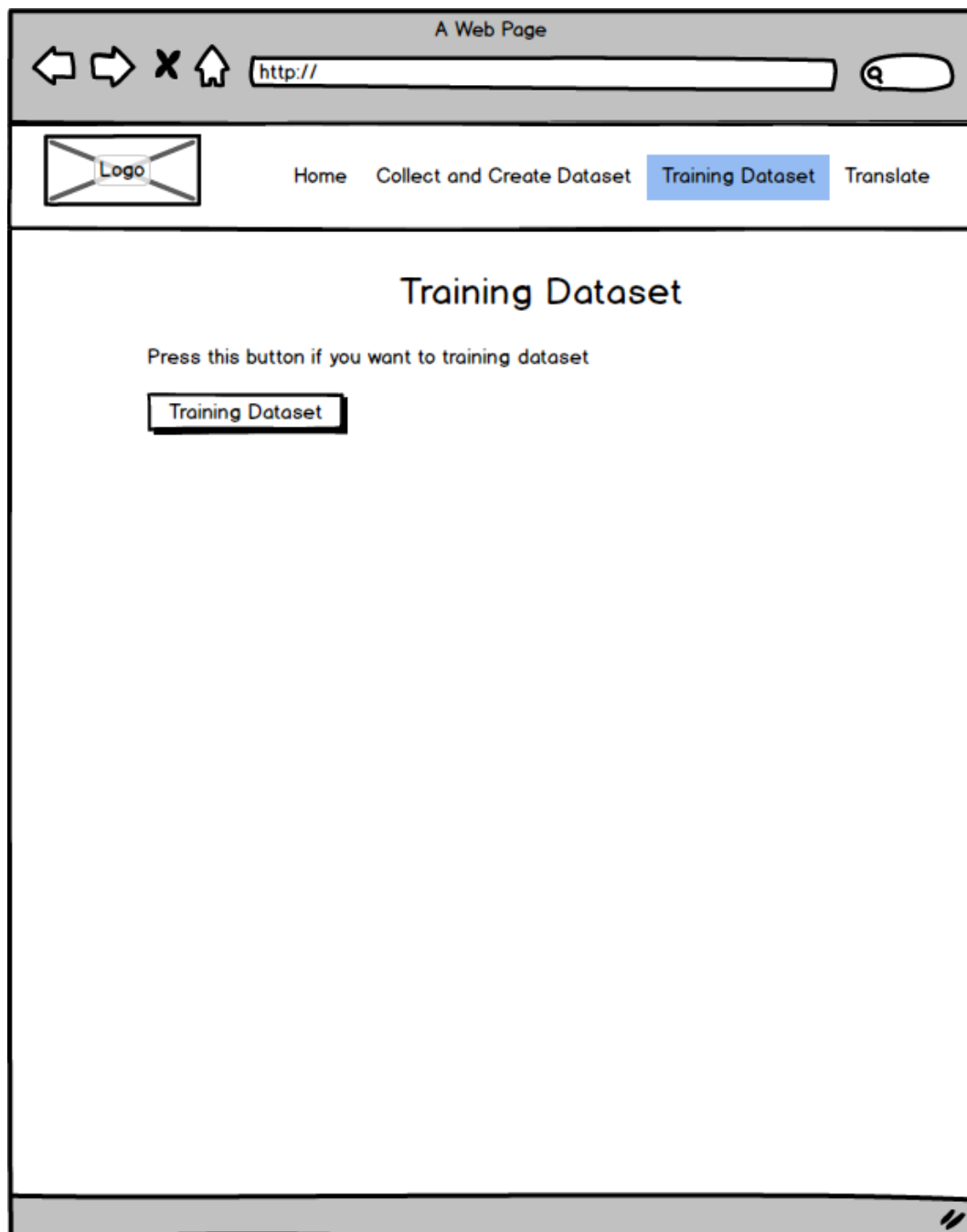
Halaman ini berupa sebuah *webcam* yang langsung menyala dimana aplikasi akan secara otomatis melakukan *screenshoot* sebanyak 100 kali untuk mengumpulkan *dataset* dan menyimpannya ke *folder* aplikasi untuk di-*create* dan di-*training*.



Gambar 3.8 Rancangan Halaman *Collect Dataset Using Webcam*

4. Rancangan Halaman *Training Dataset*.

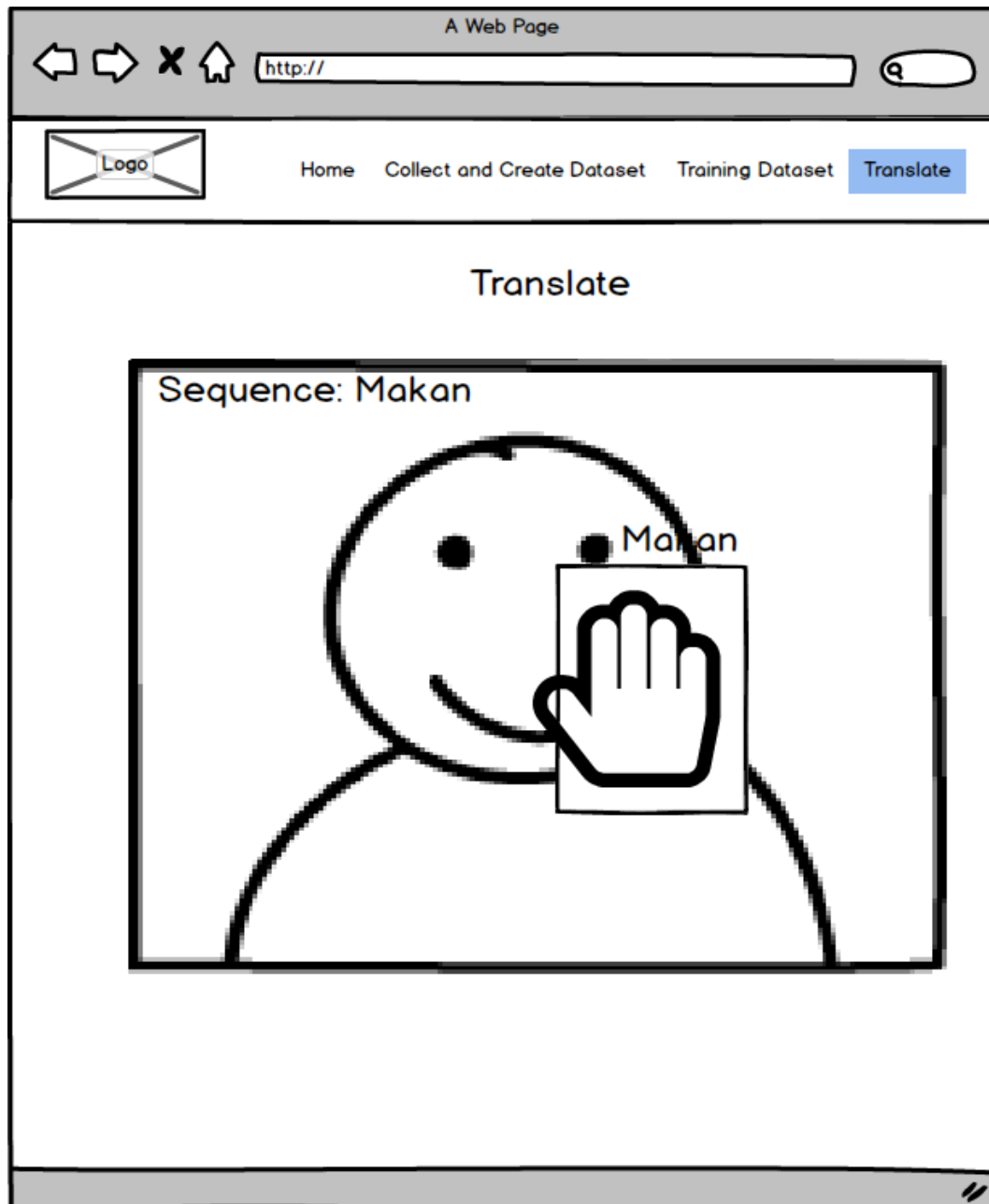
Halaman yang berisikan sebuah tombol untuk melakukan proses *training dataset* untuk membuat model klasifikasi penerjemah bahasa isyarat dengan menggunakan algoritma *Random Forest Classifier*.



Gambar 3.9 Rancangan Halaman *Training Dataset*

5. Rancangan Halaman Penerjemahan Bahasa Isyarat.

Halaman yang berisikan fitur penerjemahan bahasa isyarat secara *real time* melalui *webcam* sistem. Pada tampilan ini setiap gerakan bahasa isyarat akan dideteksi dengan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* kemudian diterjemahkan menjadi teks.



Gambar 3.10 Rancangan Halaman Penerjemahan Bahasa Isyarat

BAB IV

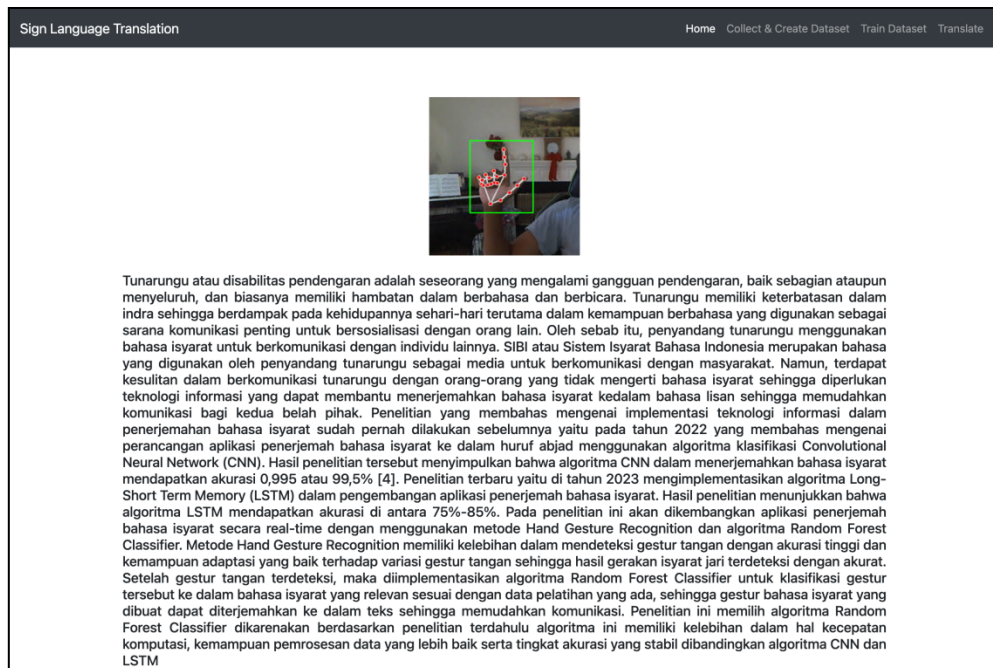
HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil penelitian ini berupa dibangunnya sebuah aplikasi penerjemah bahasa isyarat menggunakan metode *Hand Gesture Recognition* dan Algoritma *Random Forest Classifier*. Hasil aplikasi berupa aplikasi berbasis *website* dengan keseluruhan rancangannya sebagai berikut:

1. Halaman Awal.

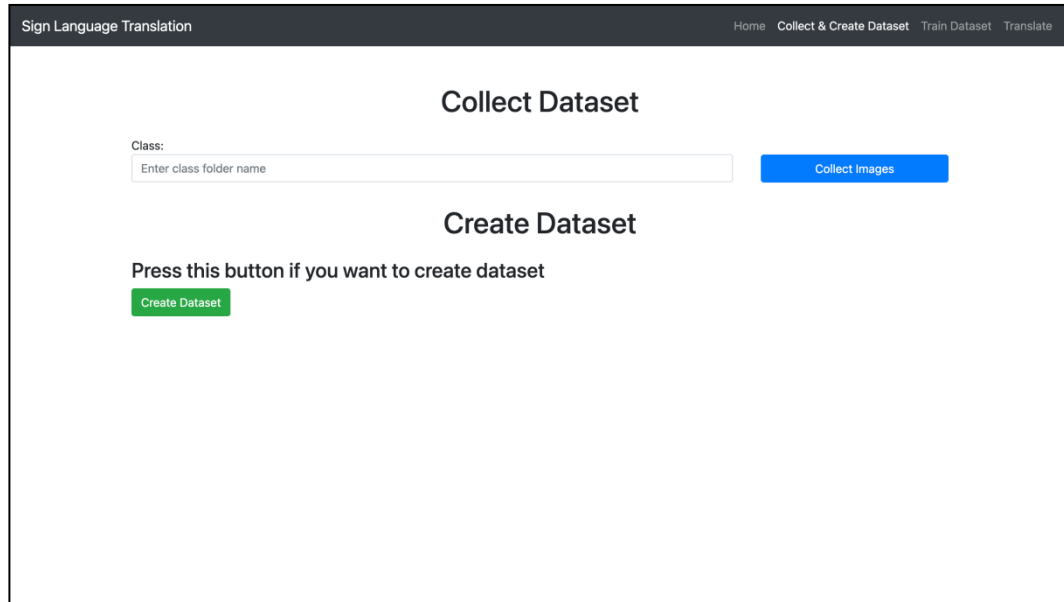
Halaman awal berisikan menu-menu navigasi aplikasi serta pengenalan mengenai model rancangan aplikasi serta penjelasannya sehingga memberikan pemahaman kepada pengguna aplikasi.



Gambar 4.1 Halaman Awal

2. Halaman *Collect and Create Dataset*.

Halaman *collect and create dataset* berisikan sebuah *form* untuk mengumpulkan *dataset* serta sebuah *button* untuk membuat *dataset* yaitu membuat *landmarks gesture* tangan dengan metode *Hand Gesture Recognition* terhadap *dataset* yang terkumpul.

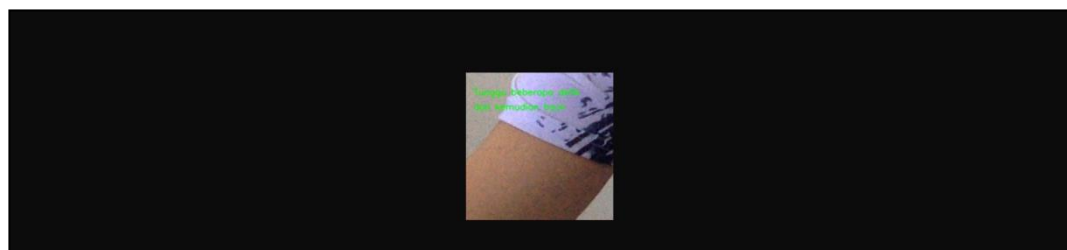


The screenshot shows a web application titled "Sign Language Translation". The navigation bar includes links for "Home", "Collect & Create Dataset", "Train Dataset", and "Translate". The main content area is divided into two sections. The top section, "Collect Dataset", features a text input field labeled "Class:" with the placeholder "Enter class folder name" and a blue button labeled "Collect Images". The bottom section, "Create Dataset", contains the instruction "Press this button if you want to create dataset" and a green button labeled "Create Dataset".

Gambar 4.2 Halaman *Collect and Create Dataset*

3. Halaman *Collect Dataset Using Webcam*.

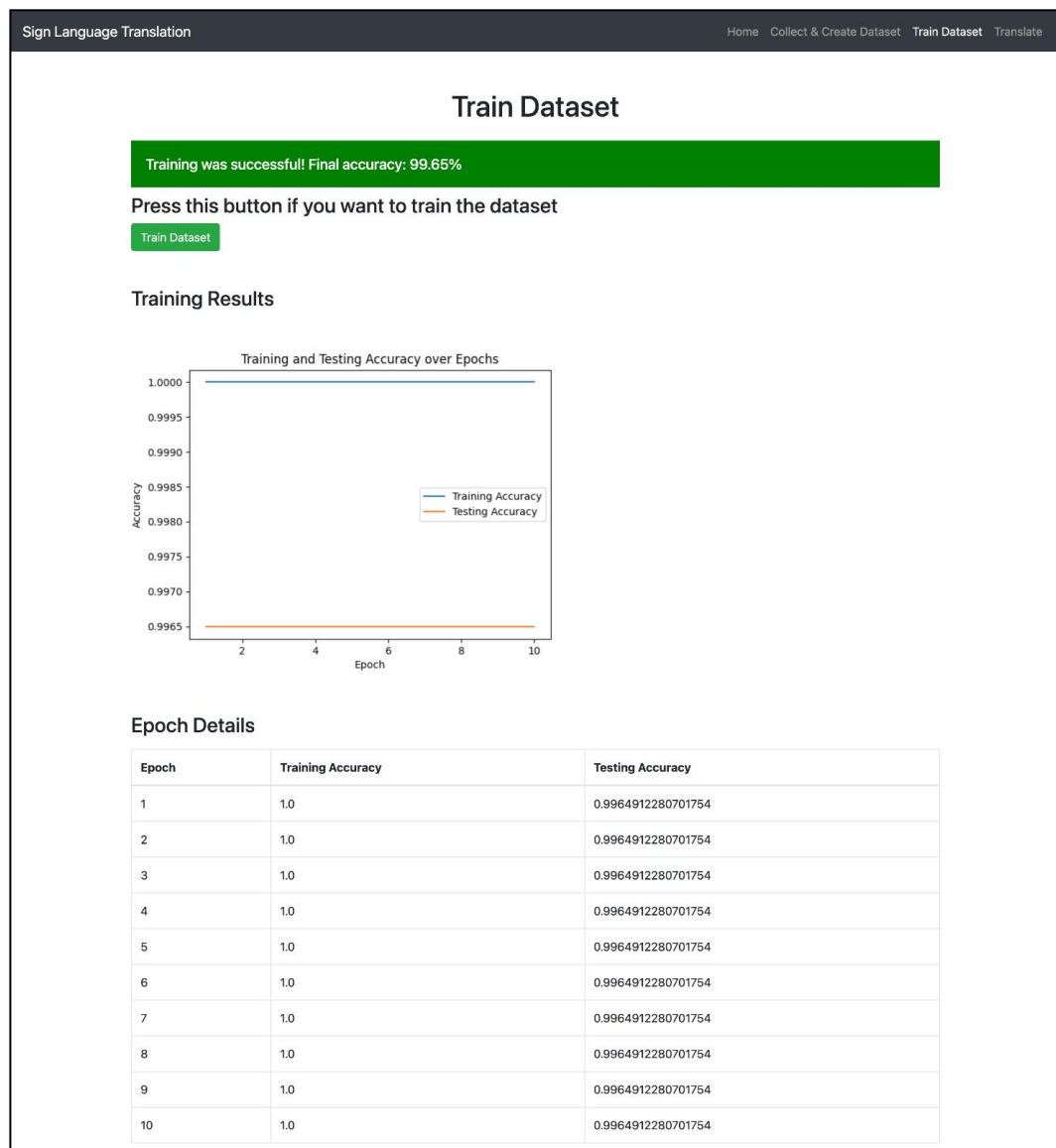
Halaman ini berupa sebuah *webcam* yang langsung menyala dimana aplikasi akan secara otomatis melakukan *screenshot* sebanyak 100 kali untuk mengumpulkan *dataset* dan menyimpannya ke *folder* aplikasi untuk di-*create* dan di-*training*.



Gambar 4.3 Halaman *Collect Dataset Using Webcam*

4. Halaman *Training Dataset*.

Halaman yang berisikan sebuah tombol untuk melakukan proses *training dataset* untuk membuat model klasifikasi penerjemah bahasa isyarat dengan menggunakan algoritma *Random Forest Classifier*.

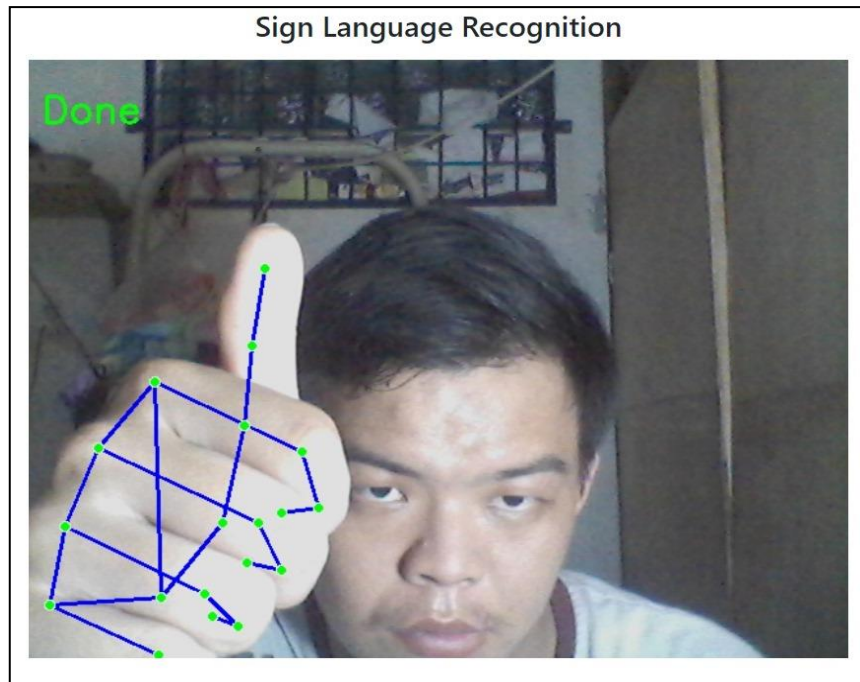


Gambar 4.4 Halaman *Training Dataset*

5. Halaman Penerjemahan Bahasa Isyarat.

Halaman yang berisikan fitur penerjemahan bahasa isyarat secara *real time* melalui *webcam* sistem. Pada tampilan ini setiap gerakan bahasa isyarat akan

dideteksi dengan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* kemudian diterjemahkan menjadi teks.

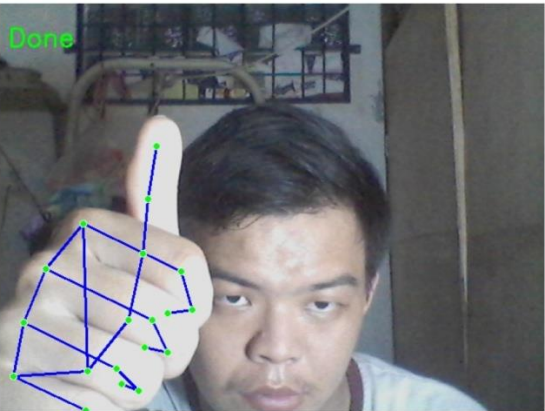
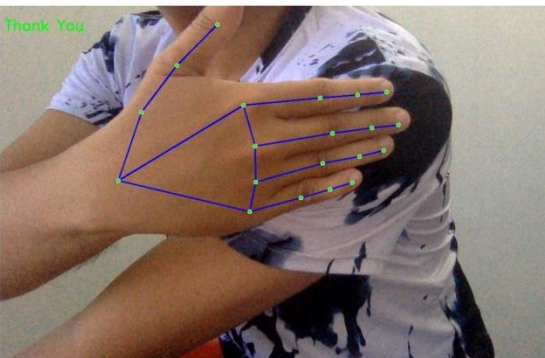
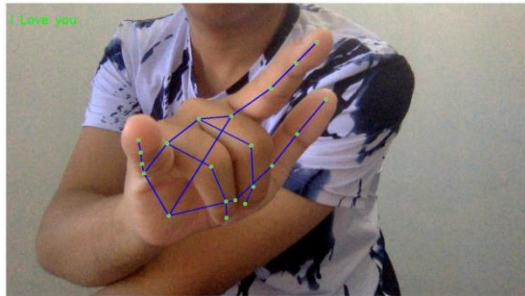


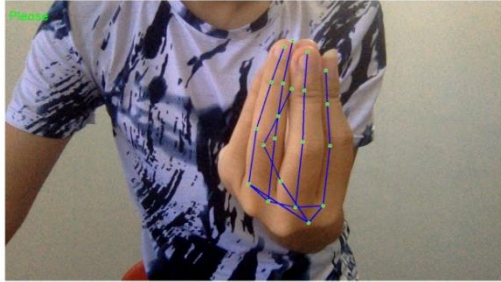
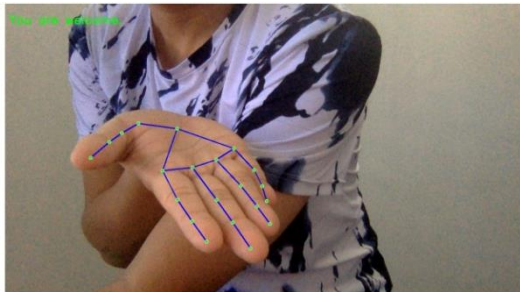
Gambar 4.5 Halaman Penerjemahan Bahasa Isyarat

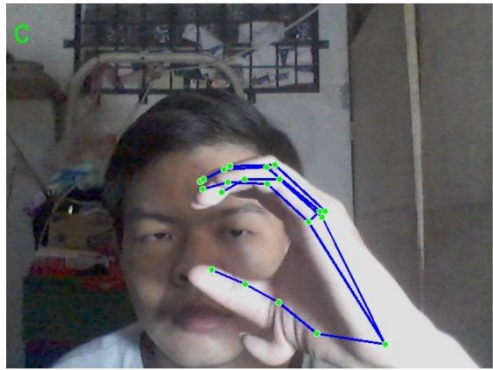
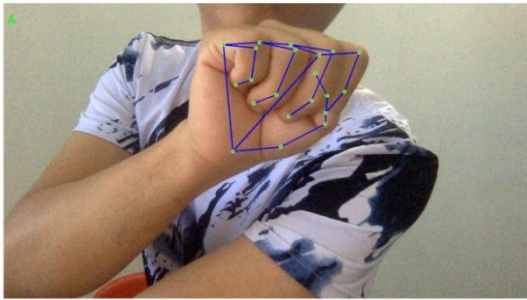
4.2 Pembahasan

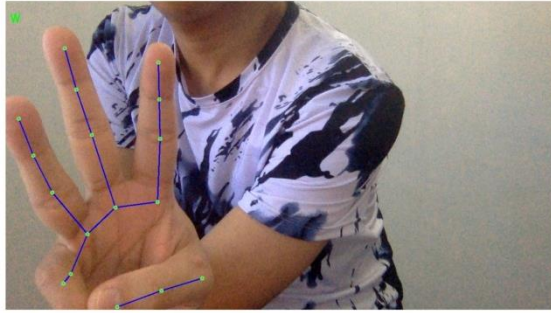
Pada sub bab ini dilakukan pembahasan terkait hasil penelitian yang dilakukan dengan melakukan pengujian aplikasi dalam menerjemahkan bahasa isyarat. Skenario pengujian dilakukan dengan pengujian kata dan pengujian huruf terhadap 6 huruf secara acak. Berikut ini, Tabel 4.1 menunjukkan skenario hasil pengujian yang dilakukan.

Tabel 4.1 Hasil Pengujian

Skenario	Hasil Pengujian	Hasil	Kesimpulan
Isyarat Kata "Hello"	Sign Language Recognition 	W	Gagal
Isyarat Kata "Done"	Sign Language Recognition 	Done	Berhasil
Isyarat Kata "Thank You"	Sign Language Recognition 	Thank You	Berhasil
Isyarat Kata "I Love You"	Sign Language Recognition 	I Love You	Berhasil

Isyarat Kata "Please"	Sign Language Recognition 	Please	Berhasil
Isyarat Kata "Sorry"	Sign Language Recognition 	Sorry	Berhasil
Isyarat Kata "You Are Welcome"	Sign Language Recognition 	You Are Welco me	Berhasil
Isyarat Huruf "A"	Sign Language Recognition 	A	Berhasil

Isyarat Huruf "B"	Sign Language Recognition 	B	Berhasil
Isyarat Huruf "C"	Sign Language Recognition 	C	Berhasil
Isyarat Huruf "E"	Sign Language Recognition 	A	Gagal
Isyarat Huruf "I"	Sign Language Recognition 	I	Berhasil

Isyarat Huruf "W"	Sign Language Recognition 	W	Berhasil
-------------------	---	---	----------

$$\begin{aligned}
 \text{Persentase Akurasi} &= (\text{Jumlah Berhasil} / \text{Total Pengujian}) * 100\% \\
 &= (11/13) * 100\% \\
 &= (0,84) * 100\% \\
 &= 84\%
 \end{aligned}$$

Pengujian dilakukan dalam dua skenario yaitu pengujian kata dan pengujian huruf. Dari tabel hasil pengujian, terlihat bahwa aplikasi berhasil menerjemahkan sebagian besar isyarat dengan tingkat akurasi 84%, berdasarkan 11 keberhasilan dari 13 pengujian.

Pada pengujian isyarat kata, aplikasi berhasil mengenali dan menerjemahkan kata-kata seperti "*Done*", "*Thank You*", "*I Love You*", "*Please*", "*Sorry*", dan "*You Are Welcome*" dengan baik. Namun, terjadi kegagalan pada isyarat kata "*Hello*", di mana aplikasi menerjemahkan isyarat tersebut sebagai "W". Hal ini kemungkinan disebabkan oleh kemiripan gerakan tangan untuk huruf "W" dan isyarat "*Hello*", yang dapat mengakibatkan kesalahan interpretasi oleh model aplikasi.

Untuk pengujian isyarat huruf, aplikasi berhasil menerjemahkan sebagian besar huruf dengan tepat, seperti "A", "B", "C", "I", dan "W". Namun, kegagalan terjadi pada huruf "E", yang diterjemahkan sebagai "A". Kemiripan bentuk isyarat antara huruf-huruf tersebut mungkin menjadi penyebab utama kegagalan ini.

Secara keseluruhan, meskipun ada beberapa kegagalan yang disebabkan oleh kemiripan isyarat, aplikasi ini menunjukkan kemampuan yang cukup baik dalam menerjemahkan bahasa isyarat dengan tingkat akurasi yang tinggi. Hasil ini menunjukkan bahwa aplikasi memiliki potensi untuk membantu komunikasi bagi pengguna bahasa isyarat, meskipun masih diperlukan beberapa perbaikan untuk meningkatkan akurasi dalam mengenali isyarat-isyarat yang mirip.

Berikutnya dilakukan pengujian *epoch* sebanyak 10 kali untuk menguji akurasi dari model yang diusulkan pada penelitian ini. Hasil pengujian *epoch* ditunjukkan pada Gambar 4.6 berikut.



Gambar 4.6 Grafik Hasil Pengujian

Selanjutnya, Tabel 4.2 menunjukkan hasil pengujian *epoch* dari model pada penelitian ini.

Tabel 4.2 Hasil Pengujian Epoch

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Testing Accuracy</i>
1	1.0	0.9988304093567252
2	1.0	0.9988304093567252
3	1.0	0.9988304093567252

4	1.0	0.9988304093567252
5	1.0	0.9988304093567252
6	1.0	0.9988304093567252
7	1.0	0.9988304093567252
8	1.0	0.9988304093567252
9	1.0	0.9988304093567252
10	1.0	0.9988304093567252

Berdasarkan Gambar 4.6 dan Tabel 4.2, dapat dilihat bahwa kinerja model dalam menerjemahkan bahasa isyarat adalah sebesar 99,88%. Terdapat beberapa alasan mengapa model yang dilatih untuk menerjemahkan bahasa isyarat bisa mencapai akurasi tinggi selama pelatihan (misalnya 99% setelah beberapa *epoch*), tetapi mengalami penurunan akurasi yang signifikan saat diuji langsung melalui *webcam*, yaitu:

1. *Overfitting*.

Model mungkin telah belajar terlalu baik pada data pelatihan sehingga tidak dapat digeneralisasi dengan baik pada data baru. Ini berarti model mengenali pola-pola spesifik dari data pelatihan tetapi tidak mampu mengenali variasi dalam data yang belum pernah dilihat sebelumnya (data dari *webcam*).

2. Perbedaan antara data pelatihan dan data uji langsung dari *webcam*.

Data yang digunakan selama pelatihan mungkin memiliki kualitas atau karakteristik yang berbeda dibandingkan dengan data dari *webcam*. Misalnya, pencahayaan, latar belakang, resolusi, sudut kamera, dan pergerakan tangan bisa berbeda, yang mengakibatkan model kesulitan untuk mengenali bahasa isyarat dari *webcam*.

3. Kurangnya data pelatihan yang representatif.

Model mungkin dilatih dengan data yang tidak cukup representatif untuk semua variasi isyarat yang mungkin muncul. Jika data pelatihan tidak

mencakup berbagai kondisi pencahayaan, sudut pandang, dan pengguna dengan karakteristik fisik yang berbeda, model tidak akan mampu mengenali isyarat dengan baik di situasi yang berbeda.

4. *Latency dan Real-time Processing Issues.*

Saat menggunakan *webcam*, ada kemungkinan *latency* (keterlambatan) dalam menangkap dan memproses gambar secara *real-time*. Hal ini bisa mempengaruhi kinerja model, terutama jika model membutuhkan waktu pemrosesan yang cukup lama untuk setiap *frame*.

5. Perbedaan resolusi dan *frame rate*.

Jika resolusi dan *frame rate* dari *webcam* berbeda dengan yang digunakan selama pelatihan, model bisa mengalami kesulitan dalam mengenali isyarat dengan benar. Misalnya, *frame rate* yang lebih rendah bisa menyebabkan kehilangan detail gerakan.

Selanjutnya, diuraikan kelebihan dan kekurangan dari aplikasi penerjemah bahasa isyarat menggunakan metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* yang dibangun. Kelebihan dari aplikasi yang dibangun yaitu:

1. Aplikasi yang dibangun berhasil menerjemahkan bahasa isyarat ke dalam teks secara *real-time*, sehingga memudahkan komunikasi bagi penyandang disabilitas tunarungu dengan orang-orang yang tidak mengerti bahasa isyarat.
2. Aplikasi ini berhasil menunjukkan akurasi 84% dalam menerjemahkan *gesture* bahasa isyarat ke dalam teks. Metode *Hand Gesture Recognition* yang digunakan terbukti efektif dalam mendeteksi gestur tangan dengan akurasi

tinggi. Selain itu, algoritma *Random Forest Classifier* memberikan hasil yang stabil dalam klasifikasi gestur.

Kekurangan dari aplikasi yang dibangun yaitu:

1. Dataset pada penelitian ini diambil dari dataset lokal dengan setiap kata bahasa isyarat masing-masing memiliki 100 gambar data latih. Hal ini membatasi variasi data yang dapat digunakan untuk melatih model, yang mungkin berdampak pada akurasi dan kemampuan aplikasi dalam menghadapi variasi gestur yang lebih luas.
2. *Gesture* bahasa isyarat yang dapat diterjemahkan oleh aplikasi ini hanya dibatasi pada huruf alphabet dari A hingga C, serta beberapa kata seperti *hello*, *done*, *thank you*, *i love you*, *sorry*, *please*, dan *you are welcome*. Batasan ini membatasi penggunaan aplikasi untuk komunikasi yang lebih kompleks dan beragam.
3. *Output* hasil terjemahan hanya disajikan dalam bentuk papan teks yang dapat dilihat di *WebCam*. Hal ini mungkin membatasi kenyamanan dan fleksibilitas pengguna dalam melihat hasil terjemahan, terutama jika mereka ingin menggunakan aplikasi ini dalam situasi komunikasi yang lebih dinamis dan beragam..
4. Aplikasi yang dibangun berbasis *website* dan bersifat *real-time*. Meskipun ini memberikan kemudahan akses, namun mungkin terdapat keterbatasan dalam hal performa dan keandalan jika diakses melalui perangkat dengan koneksi internet yang kurang stabil atau perangkat keras yang tidak mendukung.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan yang diuraikan berkaitan dengan hasil penelitian yang didapatkan yaitu:

1. Aplikasi yang dibangun berhasil memfasilitasi komunikasi bagi penyandang disabilitas tunarungu dengan menerjemahkan bahasa isyarat ke dalam teks secara *real-time*.
2. Berdasarkan pengujian yang dilakukan, aplikasi ini mencapai akurasi 84% dalam menerjemahkan *gesture* bahasa isyarat ke dalam teks secara *real-time*. Metode *Hand Gesture Recognition* dan algoritma *Random Forest Classifier* terbukti efektif dalam mendeteksi dan mengklasifikasikan gestur tangan.

5.2 Saran

Adapun saran yang diuraikan berkaitan dengan hasil penelitian yang didapatkan yaitu:

1. Untuk meningkatkan akurasi dan kemampuan aplikasi dalam menghadapi variasi gestur yang lebih luas, disarankan untuk memperluas dataset yang digunakan dalam pelatihan model. Mengumpulkan lebih banyak data dari berbagai sumber dapat membantu meningkatkan kualitas model yang dihasilkan.

2. Aplikasi ini saat ini hanya dapat menerjemahkan beberapa huruf dan kata tertentu. Untuk meningkatkan kegunaan aplikasi, disarankan untuk menambah jumlah gesture yang dapat diterjemahkan, termasuk lebih banyak huruf, kata, dan frasa yang sering digunakan dalam komunikasi sehari-hari.
3. Disarankan untuk mengembangkan *output* hasil terjemahan agar tidak hanya terbatas pada papan teks di *WebCam*. Misalnya, menambahkan fitur *audio* atau teks yang dapat disalin dapat meningkatkan kenyamanan dan fleksibilitas pengguna dalam melihat dan menggunakan hasil terjemahan.
4. Untuk memastikan performa dan keandalan yang lebih baik, disarankan untuk mengoptimalkan aplikasi berbasis *website* ini agar dapat diakses melalui perangkat dengan koneksi internet yang kurang stabil atau perangkat keras yang tidak mendukung. Pengembangan aplikasi versi *mobile* atau *desktop* mungkin juga menjadi pertimbangan untuk meningkatkan aksesibilitas dan penggunaan aplikasi.

DAFTAR PUSTAKA

- [1] S. M. Ulfah and S. Ubaidah, “Penerapan Bahasa Isyarat dalam Pembelajaran bagi Anak Berkebutuhan Khusus Tuna Rungu,” *J. Disabil. Stud. Res.*, vol. 2, no. 1, pp. 06–23, 2023.
- [2] R. Haris Alfikri, M. Siswo Utomo, H. Februariyanti, and E. Nurwahyudi, “Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN Berbasis Android,” *J. Teknoinfo*, vol. 16, no. 2, pp. 183–197, 2022, [Online]. Available: <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>.
- [3] S. Nur, A. N. Assyifa, and H. Nurjannah, “Pengembangan Aplikasi Penerjemah Bahasa Isyarat Indonesia (BISINDO) Menggunakan Metode Long-Short Term Memory,” *Edusaintek J. Pendidikan, Sains dan Teknol.*, vol. 11, no. 1, pp. 13–30, 2024.
- [4] M. Yunus and Y. Anwar, “Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke Dalam Huruf Abjad,” *J. Sintaks Log.*, vol. 2, no. 1, pp. 257–262, 2022, doi: 10.31850/jsilog.v2i1.1726.
- [5] T. He, C. Xie, Q. Liu, S. Guan, and G. Liu, “Evaluation and Comparison of Random Forest and A-LSTM Networks for Large-scale Winter Wheat Identification,” *Mdpi*, vol. 11, no. 14, pp. 1–21, 2019.
- [6] V. R. Kakularam and L. R. Parvathy, “An Innovative Analysis of Predicting Melanoma Skin Cancer Using CNN and Random Forest Algorithm,” *Adv. Parallel Comput.*, pp. 352–359, 2022, doi: 10.3233/APC220049.
- [7] F. D. Marleny, *Pengolahan Citra Digital Menggunakan Python*. Jawa Tengah: CV. Pena Persada, 2021.
- [8] Muttaqin *et al.*, *Implementasi Artificial Intelligence (AI) Dalam Kehidupan*. Medan: Yayasan Kita Menulis, 2023.
- [9] A. A. Permana *et al.*, *Machine Learning*. Padang: PT Global Eksekutif Teknologi, 2022.
- [10] Amna *et al.*, *Data Mining*. Padang: PT Global Eksekutif Teknologi, 2023.

- [11] S. S. Makahaube, A. M. Sambul, and S. R. Sompie, "Implementation of Gesture Recognition Technology for Automated Education Service Kiosk," *J. Tek. Inform.*, vol. 16, no. 4, pp. 1–8, 2021.
- [12] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Front. Psychol.*, vol. 8, no. 10, pp. 1–12, 2017, doi: 10.3389/fpsyg.2017.01745.
- [13] B. P. Hartato, "Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [14] F. Diba, M. S. Lydia, and P. Sihombing, "Analisis Random Forest Menggunakan Principal Component Analysis Pada Data Berdimensi Tinggi," *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2152–2160, 2023, doi: 10.33022/ijcs.v12i4.3329.

DAFTAR RIWAYAT HIDUP

Nama : Andrian Anjaya
Tempat/Tanggal Lahir : Medan, 5 September 2002
Jenis Kelamin : Laki - Laki
Agama : Buddha
Alamat : Jl.Ilyas No.6B Simpang kantor Medan-Labuhan
Email : andriananjaya05@gmail.com

Pendidikan Formal :

2007 - 2014 : SD Dr.Wahidin Sudirohusodo
2014 - 2017 : SMP Dr.Wahidin Sudirohusodo
2017 - 2020 : SMA Dr.Wahidin Sudirohusodo
2020 - 2024 : S1 Teknik Informatika STMIK TIME Medan

Pengalaman Kerja :

Seprtember 2020 – Sekarang : Staff Administrasi Gudang PT. United Rope

Medan, 30 Juli 2022

Andrian Anjaya



SURAT KEPUTUSAN
NO. 065/SK/P1/STMIKTIME/XII/2023

Tentang
PENGANGKATAN DOSEN PEMBIMBING SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME

1. **Membaca** : a. Hasil pemeriksaan berkas dan kelengkapan mahasiswa.
b. Presentasi Proposal Judul Skripsi mahasiswa tertanggal 09 Oktober – 18 November 2023.
2. **Mengingat** : a. Kurikulum Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time Program Studi Teknik Informatika.
b. Penyesuaian teknis pelaksanaan Skripsi Program Studi Teknik Informatika.
3. **Menimbang** : Bahwa skripsi mahasiswa merupakan wujud aplikasi kristalisasi sikap kemampuan dan ketrampilan mahasiswa/i dalam upaya menerapkan ilmu pengetahuan yang telah dipelajari dibangku kuliah, serta skripsi merupakan salah satu mata kuliah yang memiliki bobot 6 SKS, maka dipandang perlu untuk menetapkan Dosen Pembimbing I dan Dosen Pembimbing II Skripsi.

4. **Menetapkan** :
Pertama :

Mengangkat Saudara :

Pembimbing I	Robet, M.Kom
Pembimbing II	Chandra, M.Kom

Sebagai Dosen Pembimbing Skripsi atas :

2044026	Andrian Anjaya	Teknik Informatika
---------	----------------	--------------------

Dengan Judul Skripsi :

Perancangan Aplikasi Penerjemah Bahasa Isyarat Menggunakan Metode Hand Gesture Recognition Dan Algoritma Random Forest Classifier

Mahasiswa Program Strata Satu (S1) Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time.

- Kedua : Apabila terdapat kekeliruan di kemudian hari tentang penetapan Dosen Pembimbing Skripsi ini akan diperbaiki sebagaimana mestinya
- Ketiga : Surat Keputusan ini berlaku sejak tanggal ditetapkan 07 Desember 2023.

Ditetapkan di
Medan, 07 Desember 2023



Hendri, M. Kom
Pembantu Ketua I STMIK TIME

Jl. Merbabu No.32 AA - BB Medan
+6261-4561932 | +6261-4533681 (Fax)
www.stmik-time.ac.id | elearning.stmik-time.ac.id
stmiktime@gmail.com





STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : ANDRIAN ANJAYA
NIM : 2044026
JUDUL SKRIPSI : PERANCANGAN APLIKASI PENERJEMAH BAHASA ISYARAT
MENGUNAKAN METODE HAND GESTURE RECOGNITION DAN
ALGORITMA RANDOM FOREST CLASSIFIER
NAMA DOSEN : ROBET, M.KOM
DOSEN
PEMBIMBING : ☒ I (SATU) ☐ II (DUA)

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
27 Desember 2023	REVISI RUMUSAN MASALAH BAB I	
15 Januari 2024	ACC BAB I	
13 Maret 2024	HAND GESTURE RECOGNITION MODEL BELUM DIJELASKAN	
15 April 2024	ACC BAB II	
27 April 2024	MODEL HAND GESTURE BELUM DIJELASKAN MENGGUNAKAN METODE APA	
02 Mei 2024	ACC BAB III	
26 Juni 2024	REVISI PROGRAM UNTUK FITUR TRAINING DATASET	
16 Juli 2024	ACC PROGRAM	
20 Juli 2024	MASUKKAN HASIL TRAINING KE BAB IV	
26 Juli 2024	ACC BAB IV DAN BAB V	

Ketentuan :

1. Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
2. Formulir Bimbingan wajib diisi dan ditunjukkan ke dosen pembimbing tiap bimbingan
3. Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
4. Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan, 26 Juli 2024
Dosen Pembimbing,



ROBET, M.KOM

Mahasiswa,

ANDRIAN ANJAYA



STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : ANDRIAN ANJAYA
NIM : 2044026
JUDUL SKRIPSI : PERANCANGAN APLIKASI PENERJEMAH BAHASA ISYARAT
 MENGGUNAKAN METODE HAND GESTURE RECOGNITION & ALGORITMA
 RANDOM FOREST CLASSIFIER
NAMA DOSEN : CHANDRA, M.KOM
DOSEN : ☐ I (SATU) ☒ II (DUA)
PEMBIMBING

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
17 Januari 2024	Revisi Bab 1 Istilah asing, tipografi, rumusan masalah, Batasan masalah	
20 Januari 2024	Revisi Bab 1 Revisi belum sesuai bimbingan sebelumnya	
22 Januari 2024	Acc Bab 1	
16 April 2024	Revisi Bab 2 Tambahkan teori bahasa pemrograman Phyton dan IDLE	
25 April 2024	Acc Bab 2	
2 Mei 2024	Acc Bab 3	
26 Juli 2024	Acc Bab 4 dan 5	

Ketentuan :

- Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
- Formulir Bimbingan wajib diisi dan ditunjukkan ke dosen pembimbing tiap bimbingan
- Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
- Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan,

Dosen Pembimbing,



CHANDRA, M.KOM

Mahasiswa,

ANDRIAN ANJAYA