

BAB III METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai subjek dan objek penelitian, alat dan data yang digunakan, serta diagram alir penelitian yang dilakukan.

3.1 Subjek dan Objek Penelitian

Subjek pada penelitian ini adalah teknologi dan model yang digunakan yaitu Mediapipe dan ANN, sedangkan objek penelitiannya adalah mengaplikasikan teknologi tersebut pada pengenalan sistem isyarat bahasa Indonesia (SIBI).

3.2 Alat dan Bahan Penelitian

Bagian ini dijelaskan alat-alat dan bahan atau data yang digunakan dalam penelitian ini.

3.2.1 Alat Penelitian

Penelitian ini memakai 2 perangkat, pertama untuk proses dataset sampai model dan yang kedua untuk menguji model. Perangkat-perangkat tersebut terdiri dari laptop (fisik) dan Google Colaboratory (non-fisik). Spesifikasi ada di Tabel 3.1.

Table 3.1 Spesifikasi Perangkat yang Digunakan


Perangkat	Spesifikasi
Google Colaboratory	Processor Intel® Xeon®, 2.30 GHz, 2 cores
	Memori RAM sebesar 12 GB
	Penyimpanan Data sebesar 25 GB
	Jenis GPU Nvidia K80 12 GB
Laptop	Processor AMD Ryzen 3, 2.6 GHz, 4 cores
	Memori RAM 8 GB
	Penyimpanan data sebesar 512 GB
	Jenis GPU AMD Radeon RX Vega 3
	Kamera <i>webcam</i> resolusi 640x480 piksel

3.2.2 Bahan atau Data Penelitian

Penelitian ini menggunakan data yang diperoleh dari situs Kamus SIBI Kemendikbud berupa 21 video, serta 21 video tambahan yang direkam menggunakan kamera *webcam* penelitian. Setiap video menampilkan peragaan bahasa isyarat untuk satu kosakata. Secara keseluruhan, terdapat 42 video dengan 21 kosakata bahasa isyarat yang digunakan dalam penelitian ini, dan daftar kosakata tersebut dapat dilihat pada Tabel 3.1.

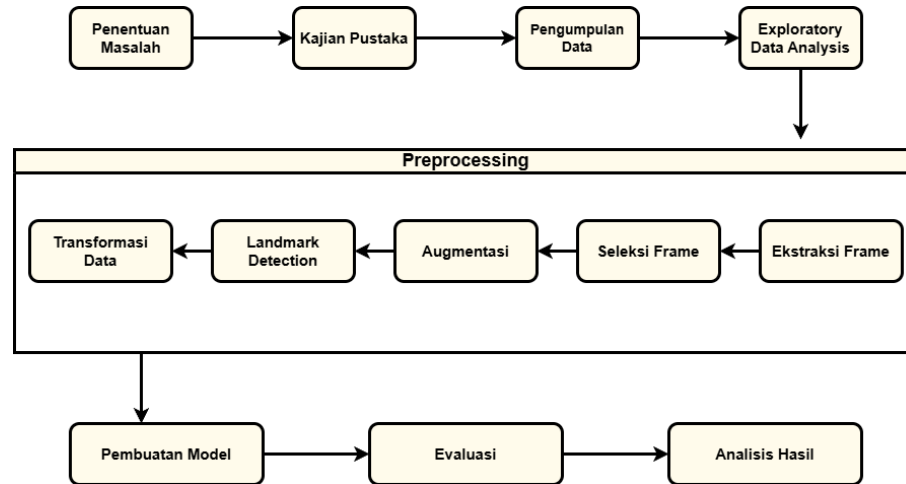
Table 3.2 Kosakata Isyarat SIBI yang Digunakan

No	Contoh Frame	Isyarat Kosakata	No	Contoh Frame	Isyarat Kosakata	No	Contoh Frame	Isyarat Kosakata
1		Apa	2		Bagaimana	3		Belum
4		Berapa	5		Dengan	6		Di
7		Dia	8		Ingin	9		Kami
10		Kamu	11		Kapan	12		Makan

No	Contoh Frame	Isyarat Kosakata	No	Contoh Frame	Isyarat Kosakata	No	Contoh Frame	Isyarat Kosakata
13		Mengapa	14		Mereka	15		Minum
16		Pergi	17		Rumah	18		Saya
19		sekolah	20		Siapa	21		Tidur

3.3 Diagram Alir Penelitian

Dalam penelitian ini terdapat diagram alir yang akan menjelaskan proses penelitian seperti gambar 3.1 berikut:



Gambar 3.1 Diagram Alir Penelitian

3.3.1 Penentuann Masalah

Penelitian dimulai dengan menentukan masalah yang diteliti berdasarkan fakta yang ada, dan juga menentukan batasan masalah sampai mana yang akan diteliti. Tahapan ini memberikan peneliti gambaran besar mengenai tujuan dan manfaat dari penelitian pengenalan bahasa isyarat SIBI.

3.3.2 Kajian Pustaka

Tahap selanjutnya dilanjutkan dengan mempelajari berbagai refrensi penelitian sebelumnya yang terkait objek dan metode yang digunakan. Topik-topik penelitian sebelumnya yang dijadikan refrensi antara lain meliputi *artificial neural network*, pengenalan pola, pendeteksian objek dan pengolahan citra digital.

3.3.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa 21 video yang diunduh dari situs Kamus SIBI Kemendikbud, serta 21 video lainnya yang direkam menggunakan kamera webcam penelitian dengan jarak dari objek antara 100 hingga 120 cm dan rata-rata intensitas cahaya 72 lux. Total terdapat 42 video dengan 21 kosakata bahasa isyarat yang digunakan dalam penelitian ini, dan kosakata apa saja yang digunakan telah disebutkan pada bagian bahan penelitian.

3.3.4 *Exploratory Data Analysis*

Tahapan ini dilakukan untuk memahami data yang akan digunakan sebelum mengembangkan model atau melakukan analisis lebih lanjut, hal ini penting untuk menggali informasi yang relevan dari data yang digunakan. Informasi yang dicari pada tahapan ini yaitu mencari sebaran warna piksel dalam video dengan mengambil satu frame dari setiap label lalu dihitung distribusi piksel warnanya dan ditampilkan dengan diagram histogram, informasi yang didapatkan dapat memberikan wawasan tentang kondisi pewarnaan dan kualitas gambar. Selain itu, menghitung jumlah frame setiap detiknya untuk memahami durasi dan kecepatan pergerakan dalam video tersebut.

Informasi terakhir yang dicari adalah resolusi video dengan menghitung panjang dan lebar piksel frame di setiap video. Dengan memeriksa sebaran warna piksel, jumlah frame, dan resolusi video, peneliti dapat memahami karakteristik dataset dan melakukan analisis yang sesuai dengan data video yang digunakan.

3.3.5 *Preprocessing*

Pada tahapan ini terdapat lima proses yang dilakukan, yaitu ekstraksi frame, seleksi frame, augmentasi, *landmark detections*, dan transformasi data. Tujuan dari tahapan ini untuk mengolah data yang awalnya berbentuk video menjadi data tabular yang dapat digunakan oleh model.

3.3.5.1 *Ekstraksi Frame*

Data video yang telah didapatkan harus dipecah terlebih dahulu untuk menjadi citra-citra gambar yang menyusun video tersebut, tahap ini menggunakan *library* python untuk mengolah gambar dan video yang bernama OpenCV. Setiap citra dilakukan proses *cropping* sesuai dengan posisi objek yang memperagakan bahasa isyarat, ukurannya berdasarkan lebar citra awal kemudian dilakukan *resizing* ke ukuran 480 x 480 piksel. Misalnya citra yang didapatkan dari data video yang resolusinya 1920 x 1080 piksel dilakukan *cropping* dengan ukuran 1080 x 1080 piksel lalu dilanjutkan dengan melakukan *resizing* ke ukuran 480 x 480 piksel.

Proses *cropping* yang dilakukan sebelum *resizing* berguna untuk mengurangi distorsi gambar jika ukuran gambar awal berbeda dengan ukuran yang diinginkan, dan juga mempertahankan kualitas detail pada objek gambar yang digunakan.

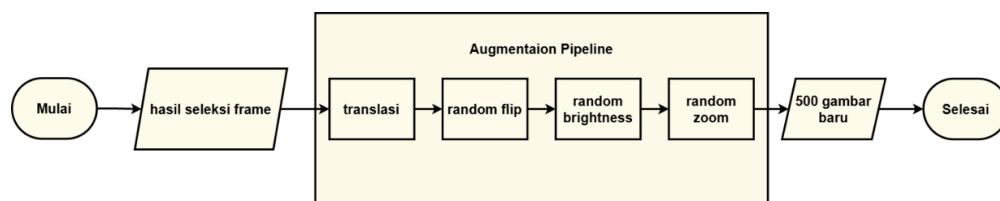
3.3.5.2 Seleksi Frame

Dari citra-citra tersebut tidak semuanya digunakan dalam penelitian, dipilih citra mana yang menunjukkan dimulainya gerakan suatu isyarat sampai gerakannya berakhir, dan citra yang dipilih juga harus menunjukkan salah satu bagian dari gerakan suatu kata isyarat yang terlihat tidak buram.

Tahap ini dikumpulkan juga citra yang menunjukkan gerakan yang bukan bahasa isyarat, kumpulan citra ini akan menjadi target bias. Semua citra tersebut dikumpulkan dan disimpan untuk dilanjutkan ke tahapan augmentasi citra.

3.3.5.3 Augmentasi

Citra gambar yang telah dipilih akan diduplikasi dan dimanipulasi dengan berbagai variasi untuk memperbanyak data, tahapan ini biasanya disebut juga dengan augmentasi.



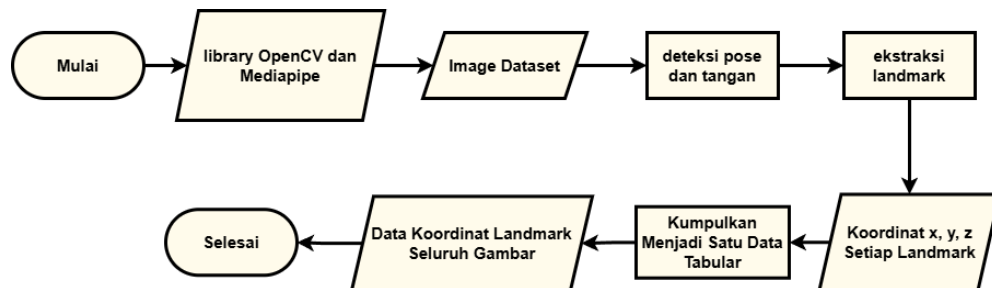
Gambar 3.2 *Flowchart* Augmentasi

Proses augmentasi dibuat dalam bentuk pipeline yang alurnya digambarkan pada *flowchart* gambar 3.2, tahap pertama dilakukan proses translasi untuk menggeser citra dari posisinya ke arah atas, bawah, kanan, atau kiri secara acak. Tetapi, pergeseran citra dibatasi pada sumbu horizontal dan vertikal sebesar -10 sampai 10 piksel. Kemudian, masuk ke tahap *flip* untuk membalikkan citra berdasarkan sumbu horizontal ke arah kiri atau kanan dengan dilakukannya secara acak. Tahap selanjutnya yaitu *random brightness* dilakukan untuk memanipulasi citra supaya memiliki intensitas cahaya yang berbeda dari citra masukannya, secara acak besaran intensitas cahaya yang digunakan berada di rentang 40% sampai 160% yang berarti akan ada citra yang lebih gelap dan lebih terang dari citra masukannya. Tahap terakhir dilakukan *random zoom* dengan skala dari 0.8 sampai 1.1 kali dari citra aslinya, yang dengan begitu akan ada citra yang dilakukan *zoom in* atau *zoom out*.

Hasil dari proses augmentasi ini mendapatkan citra gambar baru sebanyak 500 citra, hasil ini akan digabungkan dengan citra-citra yang dipilih dari proses seleksi frame berdasarkan kelas katanya. Misalnya hasil seleksi frame dari data video kosakata “Aku” mendapatkan 10 citra, kemudian citra tersebut digabung dengan 500 citra baru hasil augmentasi yang totalnya menjadi 510 citra gambar kelas kosakata “Aku” .

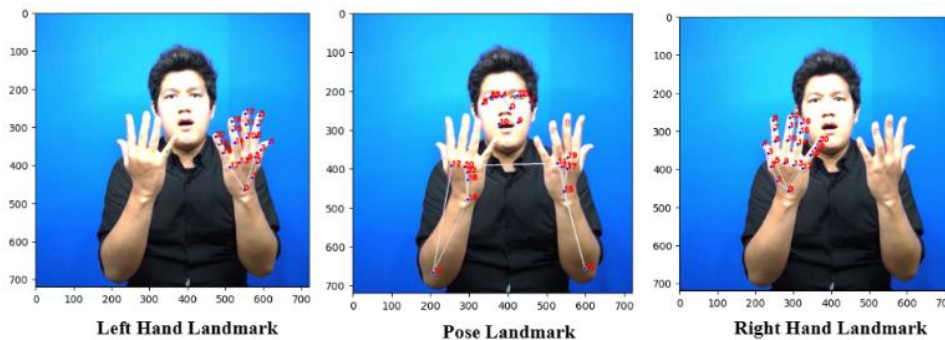
3.3.5.4 Landmark Detections

Data citra yang telah didapatkan dari tahap augmentasi dilanjutkan ke tahap *landmark detections* yang bertujuan untuk mengekstrak fitur yang terdapat di citra gambar tersebut, ada beberapa proses yang dilakukan menggunakan Mediapipe seperti pada flowchart gambar 3.3. Proses ini mendapatkan fitur yang berupa data koordinat letak titik-titik *landmark* setiap citra gambar yang terdeteksi oleh Mediapipe, koordinat *landmark* memiliki 3 titik yaitu titik x , y , dan z yang telah dinormalisasi berdasarkan ukuran asli citra.



Gambar 3.3 Flowchart Deteksi Landmark

Ada dua jenis pendeteksian yang digunakan pada penelitian ini, pertama Mediapipe mendeteksi postur tubuh manusia secara keseluruhan atau *pose landmark detection*, kedua mendeteksi lengan tangan kanan dan kiri manusia atau *hand landmark detection*. Setiap data citra yang telah diekstrak akan mendapatkan 225 fitur yang kemudian dikumpulkan menjadi satu data tabular beserta label datanya. Contoh hasil dari titik di mana saja yang terdeteksi terlihat pada gambar 3.4, dan untuk nilai koordinat detailnya terdapat pada Tabel 3.2 dan 3.3.



Gambar 3.4 Lokasi *Keypoints Landmark*

Data *landmark* yang sudah berbentuk tabular dibagi menjadi data *train set* dan data *test set*, jumlah data *test set* diambil dari 30% jumlah data setiap kelasnya dan sisanya digunakan untuk data *train set*. Data *test set* ini digunakan untuk menguji dan mengevaluasi model. Sedangkan *train set*, digunakan untuk melatih model yang dibuat.

Mediapipe dapat mengekstraksi *landmark* dari *input* gambar dengan melakukan proses sebagai berikut[21]:

1. Deteksi pose

Mediapipe menggunakan model yang bernama BlazePose[32] untuk mendeteksi pose tubuh terhadap gambar atau video yang masuk, model BlazePose merupakan model yang telah dilatih dengan kumpulan data besar gambar dan video manusia yang menggunakan arsitektur *convolutional neural network* untuk mengekstrak fitur gambar dan memprediksi posisi *landmark* tubuh.

2. Deteksi *landmark*

Pose tubuh yang telah terdeteksi dilanjutkan dengan mengidentifikasi titik kunci pada tubuh manusia dalam bergerak seperti pergelangan tangan, siku, bahu, pinggul, lutut, pengelangan kaki, dan mata.

3. Perhitungan koordinat *landmark*

Titik kunci yang telah teridentifikasi dihitung koordinat x dan y yang relatif terhadap pusat gambar atau video.

4. Estimasi Pose ke 3D

Hasil perhitungan koordinat *landmark* tahap sebelumnya digunakan untuk memprediksi posisi dan orientasi tubuh manusia dalam ruang 3D yang menggunakan model GHUM[33]. Gambar atau video yang masuk ke Mediapipe dapat menghasilkan koordinat *landmark* 3D.

Table 3.3 Contoh Data Hasil *Pose Landmark Detections*

Indeks Landmark	Nama Landmark	Koordinat Landmark		Hasil Ekstraksi Landmark		
		<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>z</i>
0	nose	406	246	0.56	0.34	-0.79
1	Left eye (inner)	424	213	0.59	0.3	-0.73
2	Left eye	432	213	0.6	0.3	-0.73
3	Left eye (outer)	442	213	0.61	0.3	-0.73
4	Right eye (inner)	385	215	0.53	0.3	-0.76
5	Right eye	369	216	0.51	0.3	-0.76
6	Right eye (outer)	356	217	0.49	0.3	-0.76
7	Left ear	457	229	0.64	0.32	-0.32
8	Right ear	335	234	0.47	0.33	-0.43
9	Mouth (left)	426	282	0.59	0.39	-0.64
10	Mouth (right)	378	286	0.53	0.4	-0.67
11	Left shoulder	528	387	0.73	0.54	-0.06
12	Right shoulder	255	393	0.35	0.55	-0.25
13	Left elbow	598	657	0.83	0.91	-0.5
14	Right elbow	207	668	0.29	0.93	-0.75
15	Left wrist	544	459	0.76	0.64	-1.41
16	Right wrist	297	480	0.41	0.67	-1.62
17	Left pinky	553	401	0.77	0.56	-1.55
18	Right pinky	297	429	0.41	0.6	-1.76
19	Left index	554	372	0.77	0.52	-1.5
20	Right index	288	396	0.4	0.55	-1.72
21	Left thumb	539	396	0.75	0.55	-1.41
22	Right thumb	295	410	0.41	0.57	-1.62
23	Left hip	497	743	0.69	1.03	0.04
24	Right hip	324	753	0.45	1.05	-0.03
25	Left knee	519	1089	0.72	1.51	-0.07
26	Right knee	320	1097	0.44	1.52	0.05
27	Left ankle	508	1374	0.71	1.91	0.41
28	Right ankle	328	1388	0.46	1.93	0.46
29	Left heel	510	1408	0.71	1.96	0.42

Indeks Landmark	Nama Landmark	Koordinat Landmark		Hasil Ekstraksi Landmark		
		<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>z</i>
30	Right heel	320	1427	0.44	1.98	0.48
31	Left foot index	495	1473	0.69	2.05	-0.14
32	Right foot index	362	1475	0.5,	2.05	-0.14

Table 3.4 Contoh Data Hasil Ekstraksi *Hand Landmark Detections*

Indeks Landmark	Nama Landmark	Lengan Tangan Kanan					Lengan Tangan Kiri				
		Koordinat Landmark		Hasil Ekstraksi Landmark			Koordinat Landmark		Hasil Ekstraksi Landmark		
		x	y	x	y	z	x	y	x	y	z
0	Wrist	550	466	0.76	0.65	0	287	459	0.4	0.64	0
1	Thumb CMC	571	434	0.79	0.6	0	258	433	0.36	0.6	0
2	Thumb MCP	583	395	0.81	0.55	-0.01	241	395	0.34	0.55	-0.01
3	Thumb IP	593	361	0.82	0.5	-0.02	234	360	0.33	0.5	-0.01
4	Thumb TIP	616	343	0.86	0.48	-0.03	219	338	0.31	0.47	-0.02
5	Index finger MCP	578	380	0.8	0.53	-0.04	252	383	0.35	0.53	-0.04
6	Index finger PIP	582	327	0.81	0.46	-0.07	252	328	0.35	0.46	-0.06
7	Index finger DIP	585	296	0.81	0.41	-0.08	250	297	0.35	0.41	-0.06
8	Index finger TIP	590	271	0.82	0.38	-0.08	247	273	0.34	0.38	-0.07
9	Middle finger MCP	560	385	0.78	0.54	-0.05	276	386	0.38	0.54	-0.05
10	Middle finger PIP	556	329	0.77	0.46	-0.07	285	326	0.4	0.45	-0.06
11	Middle finger DIP	554	294	0.77	0.41	-0.08	290	291	0.4	0.41	-0.07
12	Middle finger TIP	552	266	0.77	0.37	-0.09	291	266	0.41	0.37	-0.07
13	Ring finger MCP	537	394	0.75	0.55	-0.06	301	391	0.42	0.54	-0.05
14	Ring finger PIP	525	342	0.73	0.48	-0.08	312	336	0.43	0.47	-0.07

Indeks Landmark	Nama Landmark	Lengan Tangan Kanan					Lengan Tangan Kiri				
		Koordinat Landmark		Hasil Ekstraksi Landmark			Koordinat Landmark		Hasil Ekstraksi Landmark		
		x	y	x	y	z	x	y	x	y	z
15	Ring finger DIP	519	310	0.72	0.43	-0.09	318	305	0.44	0.42	-0.08
16	Ring finger TIP	517	286	0.72	0.4	-0.09	321	281	0.45	0.39	-0.08
17	Pinky MCP	512	406	0.71	0.56	-0.07	325	399	0.45	0.55	-0.05
18	Pinky PIP	493	369	0.69	0.51	-0.08	345	364	0.48	0.51	-0.07
19	Pinky DIP	483	347	0.67	0.48	-0.09	358	346	0.5	0.48	-0.08
20	Pinky TIP	474	326	0.66	0.45	-0.09	369	328	0.51	0.46	-0.08

3.3.5.5 Transformasi Data

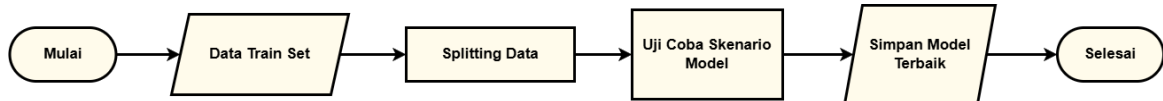
Hasil dari proses *landmark detections* belum bisa digunakan untuk pembuatan model, hal ini dikarenakan tipe data pada kolom label data *train set* dan *test set* belum bertipe numerikal. Oleh karena itu, tahapan ini akan dilakukan perubahan data di kolom tersebut yang awalnya kategorikal menjadi numerikal menggunakan teknik *label encoding* yang contohnya seperti pada Tabel 3.5.

Table 3.5 Contoh Perubahan Data Kategorikal Menjadi Numerikal

Sebelum Diubah	Setelah Diubah
A	0
B	1
C	2

3.3.6 Pembuatan Model

Tahapan berikutnya dilanjutkan dengan membangun model MLP untuk melakukan klasifikasi data koordinat *landmark* sesuai dengan kelas kosakatanya, alur proses ini seperti pada *flowchart* gambar 3.5.



Gambar 3.5 Flowchart Pembuatan Model

Data *train set* dibagi menjadi dua bagian yaitu data *training* dan data *validation* dengan persentase 70% dan 30%, kedua bagian data tersebut akan digunakan untuk proses melatih model.

Banyak literatur yang menyatakan tidak ada aturan baku dalam membuat arsitektur model ANN tipe MLP, eksperimen dan penyesuaian diperlukan untuk menemukan arsitektur yang tepat untuk masalah tertentu[34][35].

Table 3.6 Parameter *Layers*, Neuron, dan Fungsi Aktivasi

Layer		Jumlah neuron	Fungsi aktivasi
Jenis	Layer ke-		
Input	1	225	-
Hidden	2	200	ReLU
	3	175	ReLU
	4	150	ReLU
	5	125	ReLU
	6	100	ReLU

Layer		Jumlah neuron	Fungsi aktivasi
Jenis	Layer ke-		
	7	75	ReLU
Output	8	22	Softmax

Pada Tabel 3.6 menunjukkan konfigurasi parameter jumlah *layer*, neuron dan fungsi aktivasi pada arsitektur model yang digunakan di penelitian ini. Pada *hidden layer* menggunakan *layer* berlapis-lapis, ini bertujuan untuk menangkap fitur yang lebih kompleks dan abstrak[36]. Jumlah neuron di *input layer* dan *output layer* mengikuti jumlah fitur dan jumlah label targetnya, dan jumlah neuron pada *hidden layer* yang terus berkurang di setiap *layer* bertujuan supaya model dapat memadatkan informasi dan fokus pada fitur yang paling penting sehingga proses pembelajarannya lebih efisien[35]. Fungsi aktivasi ReLU pada neuron di setiap *hidden layer* dipilih karena dalam banyak literatur menggunakan fungsi ini dan dapat meningkatkan akurasi serta efisiensi pembelajaran model[37][24], neuron di *output layer* menggunakan fungsi aktivasi *softmax* dikarenakan untuk mengubah output jaringan menjadi distribusi probabilitas setiap kelasnya sehingga memudahkan untuk menginterpretasinya.

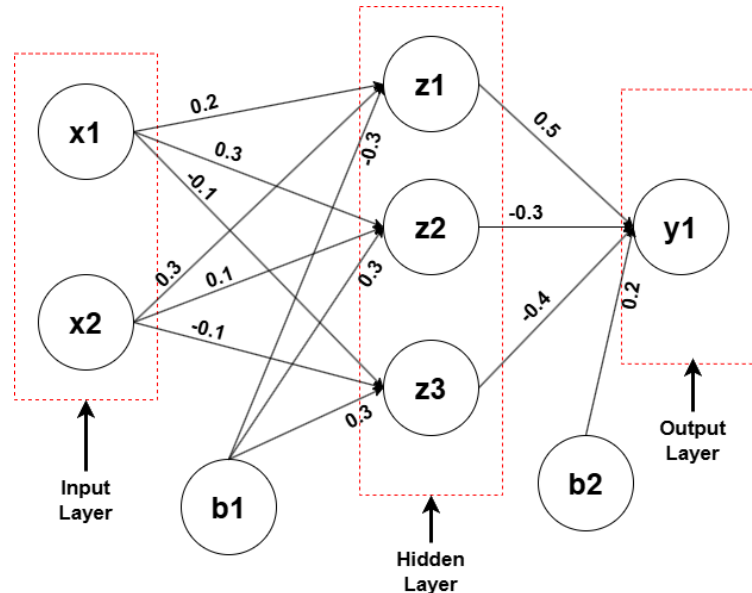
Table 3.7 Kombinasi Parameter yang Digunakan

Parameter	Nilai Parameter
<i>Learning rate</i>	0.01, 0.001, dan 0.0001
<i>Batch size</i>	16, 32, 64, dan 128
<i>epochs</i>	500 dan 100
<i>Loss function</i>	<i>Sparse categorical crossentropy</i>
<i>optimizer</i>	Adam

Pada Tabel 3.7 menunjukkan parameter *learning rate*, *batch size*, dan *epochs* lebih dari satu, ini dilakukan untuk membuat beberapa skenario uji coba model dengan kombinasi parameter tersebut. Skenario uji coba menggunakan arsitektur konfigurasi yang berbeda-beda akan diuji dengan melatihnya menggunakan data *training* dan *validation* untuk mencari konfigurasi yang dapat menghasilkan performa terbaik.

Model-model yang telah dilatih dengan konfigurasi parameter berbeda akan dipilih satu model yang skor akurasi tertinggi dan *loss* terendah, model yang telah dipilih akan disimpan untuk dilakukan evaluasi pada tahap selanjutnya. Hasil dari

tahapan ini diharapkan mendapatkan model yang mampu mengenal dan mengklasifikasikan data yang ada ke kosakata isyarat dengan akurasi terbaik.



Gambar 3.6 Ilustrasi Arsitektur Model ANN

Permisalan ada sebuah data yang memiliki 2 variabel independen x_1 , x_2 bernilai 1, 1 dan variabel dependen y yang bernilai 0 ingin diprediksi menggunakan algoritma MLP dengan ilustrasi arsitektur seperti pada gambar 3.6 ditambah dengan parameter *learning rate* sebesar 0.2 dan fungsi aktivasinya *sigmoid*.

Langkah-langkah menyelesaikannya dengan cara kerja MLP dilakukan dengan 3 tahapan utama:

1. *Feedforward*

- a) Masukkan data dari setiap neuron di *input layer* ke setiap neuron di *hidden layer* untuh mendapatkan hasil perhitungannya dengan persamaan 2.1.

$$z_1 = f(1(0.2) + 1(0.3) + (-0.3)) = f(0.2) = \frac{1}{1 + e^{-0.2}}$$

$$= 0.55$$

$$z_2 = f(1(0.3) + 1(0.1) + 0.3) = f(0.7) = \frac{1}{1 + e^{-0.7}}$$

$$= 0.67$$

$$z_3 = f(1(-0.1) + 1(-0.1) + 0.3) = f(0.1) = \frac{1}{1 + e^{-0.1}}$$

$$= 0.52$$

- b) Menghitung hasil nilai yang dikeluarkan oleh neuron y1 di *output layer*.

$$y_1 = f(0.55(0.5) + 0.67(-0.3) + 0.52(-0.4) + (-0.1))$$

$$= f(-0.24) = \frac{1}{1 + e^{0.24}} = 0.44$$

- c) Menghitung *error* antara nilai *output* dari neuron y1 dengan nilai variabel y menggunakan persamaan 2.3.

$$E = 0 - 0.44 = -0.44$$

2. Backpropagation

- a) Menghitung *error* neuron yang berada di *output layer* menggunakan persamaan 2.9.

$$\delta_o = 0.44 \times (1 - 0.44) \times (-0.44) = -0.11$$

- b) Menghitung *error* neuron yang berada di *hidden layer* menggunakan persamaan 2.10.

$$\delta_{hz1} = 0.55 \times (1 - 0.55) \times (0.5 \times (-0.11)) = -0.01$$

$$\delta_{hz2} = 0.67 \times (1 - 0.67) \times ((-0.3) \times (-0.11)) = 0.007$$

$$\delta_{hz3} = 0.52 \times (1 - 0.52) \times ((-0.4) \times (-0.11)) = 0.01$$

- c) Menghitung nilai Δw dan Δb yang terdapat diantara *hidden layer* sampai *output layer* dengan persamaan 2.5 dan 2.6.

$$\Delta w_{y1z1} = 0.2 \times (-0.11) \times 0.55 = -0.01$$

$$\Delta w_{y1z2} = 0.2 \times (-0.11) \times 0.67 = -0.01$$

$$\Delta w_{y1z3} = 0.2 \times (-0.11) \times 0.52 = -0.01$$

$$\Delta b_{12} = 0.2 \times (-0.11) = -0.02$$

- d) Menghitung nilai Δw dan Δb yang terdapat diantara *input layer* sampai *hidden layer*.

$$\Delta w_{z1x1} = 0.2 \times (-0.01) \times 1 = -0.002$$

$$\Delta w_{z2x1} = 0.2 \times 0.007 \times 1 = -0.001$$

$$\Delta w_{z3x1} = 0.2 \times 0.01 \times 1 = 0.002$$

$$\Delta w_{z1x2} = 0.2 \times (-0.01) \times 1 = -0.002$$

$$\Delta w_{z2x2} = 0.2 \times 0.007 \times 1 = -0.001$$

$$\Delta w_{z3x2} = 0.2 \times 0.01 \times 1 = 0.002$$

$$\Delta b_{11} = 0.2 \times (-0.01) = -0.002$$

$$\Delta b_{21} = 0.2 \times 0.007 = -0.001$$

$$\Delta b_{31} = 0.2 \times 0.01 = 0.002$$

3. Ubah bobot dan bias

$$w_{z1x1} = 0.2 - 0.002 = 0.198$$

$$w_{z2x1} = 0.3 - 0.001 = 0.299$$

$$w_{z3x1} = (-0.1) + 0.002 = -0.098$$

$$w_{z1x2} = 0.3 - 0.002 = 0.298$$

$$w_{z2x2} = 0.1 - 0.001 = 0.099$$

$$w_{z3x2} = (-0.1) + 0.002 = -0.098$$

$$b_{11} = (-0.3) - 0.002 = -0.302$$

$$b_{21} = 0.3 - 0.001 = 0.299$$

$$b_{31} = 0.3 + 0.002 = 0.302$$

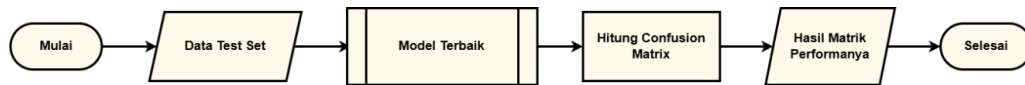
$$b_{12} = (-0.1) - 0.02 = -0.12$$

Apabila nilai *error* belum bernilai nol atau nilai *error* kurang dari batas yang ditentukan, maka ulangi tahapan ini dari awal dengan menggunakan nilai bobot dan bias yang telah diubah.

3.3.7 Evaluasi Model

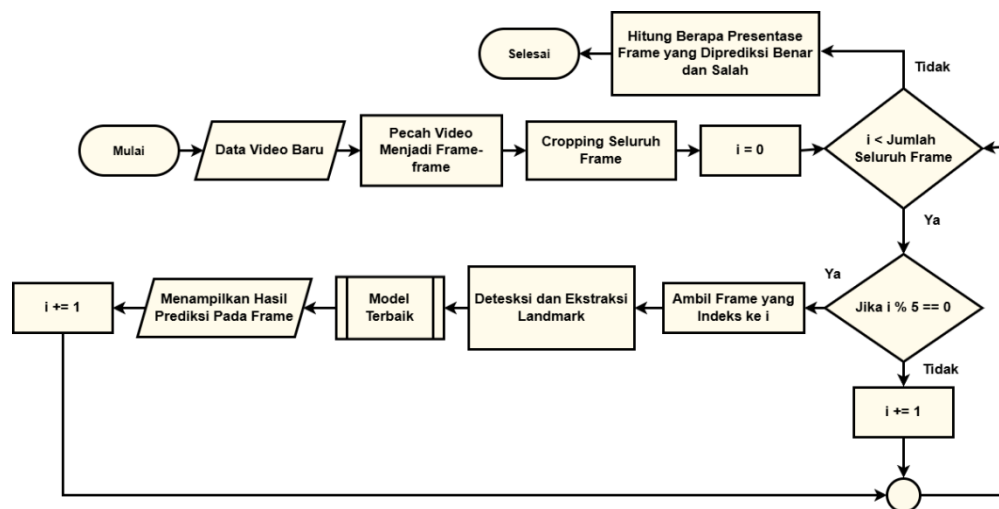
Pada tahapan ini akan dilakukan dua kali evaluasi terhadap model yang telah dibuat dari tahapan sebelumnya, evaluasi pertama dengan data *test set* dan yang kedua evaluasi dengan data video yang diambil menggunakan perangkat laptop yang spesifikasinya telah disebutkan dibagian alat penelitian.

Evaluasi pertama menggunakan data *test set* yang diambil dari hasil proses augmentasi sebanyak 30% dari masing-masing kelasnya, kemudian dimasukkan ke dalam proses evaluasi seperti pada *flowchart* gambar 3.7 untuk mendapatkan hasil *confusion matrix* dan akurasi.



Gambar 3.7 *Flowchart* Mencari Hasil *Confusion Matrix* Model

Adapun untuk evaluasi kedua yang menggunakan data video diambil menggunakan *webcam*, video tersebut dipecah menjadi banyak frame yang kemudian dilakukan *cropping* yang mengubah ukuran panjang dan lebar setiap frame menjadi 480 x 480 piksel. Selanjutnya dilakukan pendeteksian *landmark* dari frame pertama dan setiap lima frame berikutnya sampai frame terakhir, data *landmark* yang terdeteksi dimasukkan ke model untuk diprediksi dan hasilnya ditampilkan pada frame tersebut. Alur proses evaluasi ini ditunjukkan pada *flowchart* gambar 3.8.



Gambar 3.8 *Flowchart* Evaluasi Model Dengan Data Video

Setelah model yang diuji dengan seluruh data video, selanjutnya dilakukan evaluasi dengan menghitung berapa frame yang labelnya diprediksi dengan benar, hasilnya akan mendapatkan presentase jumlah frame di setiap video yang dapat diprediksi model dengan benar.

3.3.8 Analisis Hasil dan Kesimpulan

Tahapan terakhir ini akan berisi ringkasan dan temuan dari hasil penelitian yang telah dilakukan. Peneliti akan melihat bagaimana hasil dari mengimplementasikan Mediapipe dan ANN dalam mengenal kata isyarat SIBI, lalu mencatat hasilnya dan mengujinya dalam mengenal isyarat SIBI.