BAB II
TINJAUAN PUSTAKA

Bab ini menguraikan kajian pustaka serta landasan teori yang digunakan dalam penelitian ini.

2.1 Kajian Pustaka

Penelitian ini berkaitan dengan pengenalan bahasa isyarat dan gestur tubuh. Sudah banyak penelitian terdahulu yang juga mengangkat tema serupa hanya saja terdapat beberapa perbedaan, diantara perbedaannya tersebut dari segi masalah yang diangkat, algoritma yang digunakan serta sistem bahasa isyarat yang digunakan.

Penelitian terkait pengenalan abjad alfabet BISINDO sudah banyak yang dikembangkan, seperti penelitian [16] yang menggunakan metode LSTM dan CNN dalam pengenalan gambar abjad BISINDO. Data yang digunakan merupakan citra gambar abjad BISINDO yang diperagakan dan dipotret oleh peneliti itu sendiri sebanyak 1100 gambar. Hasil dari penelitian ini adalah dengan memanfaatkan gabungan metode LSTM dan CNN dapat mengenal isyarat abjad BISINDO dengan akurasi pada saat pelatihan model sebesar 90%, dan pada saat *validation* sebesar 80%. Kelemahan dari penelitian ini model hanya bisa mengenal isyarat yang diperagakan dengan tangan saja, tidak dengan gerakan wajah, mulut, dan tubuh.

Selain penelitian[16], ada juga penelitian yang menggabungkan abjad BISINDO dengan berapa kata isyarat sebagai kelas dalam sistem pengenalannya seperti pada penelitian[6]. Penelitian tersebut menggunakan metode CNN dengan dataset citra gambar berjumlah 2163 yang terdiri dari kelas abjad A-E ditambah kata “saya”, “kamu”, dan “*i love you”*. Hasil yang didapatkan dengan menggunakan dataset tersebut mampu membuat model mendapatkan perfoma akurasi yang tinggi, yaitu sebesar 99,82%.

Lalu, ada penelitian[17] yang menerapkan metode *transfer learning* dalam pengenalan isyarat abjad BISINDO, dataset yang digunakan merupakan kumpulan citra gambar isyarat abjad BISINDO dan ASL. Model dibuat dengan algoritma

CNN dan menggunakan dataset citra isyarat abjad ASL yang berjumlah 87000 citra gambar, hasil dari model tersebut memiliki performa akurasi sebesar 96%. Kemudian, model yang sudah dibuat berdasarkan data isyarat abjad ASL diterapkan *transfer learning* terhadap dataset citra isyarat abjad BISINDO yang berjumlah 2659 citra gambar. Hasil yang didapatkan dari penelitian ini menunjukkan model mengalami penurunan performa akurasi menjadi 30%, hal ini disebabkan karena model hanya bisa mengenali abjad BISINDO yang mirip dengan abjad isyarat ASL.

Penggunaan Mediapipe dalam pengembangan suatu sistem pengenalan juga sering digunakan, seperti pada penelitian[18] yang menggunakan Mediapipe dalam pengenalan gestur jari untuk mengontrol volume media suara pada perangkat komputer atau laptop secara real time. Mediapipe digunakan untuk mendeteksi letak dari ujung jari telunjuk dan ibu jari, dengan begitu jarak antar dua ujung tersebut bisa diketahui. Jarak kedua ujung dua jari tersebut yang digunakan untuk menentukan pengaturan volume media suaranya, semakin dekat jaraknya semakin kecil begitu pun sebaliknya. Tetapi, penelitian ini memiliki kelamahan yang dimana tidak bisa mengenali gestur tangan secara akurat jika teralu dekat dengan kamera webcam.

Mediapipe juga digunakan pada penelitian tentang pengenalan bahasa isyarat SIBI, seperti pada penelitian[19] yang menggunakan Mediapipe, *random forest* dan *multinomial logistic regression*. Dataset yang digunakan merupakan data primer yang berisi 1000 citra gambar alfabet dalam isyarat SIBI yang memiliki gestur statis, alfabet J dan Z tidak dimasukkan karena kedua alfabet ini memiliki gestur dinamis. Mediapipe digunakan untuk mendeteksi 21 titik yang berada di telapak tangan di setiap gambar dataset, koordinat 21 titik ini yang dikumpulkan dan digunakan untuk pembuatan model. *Random forest* dan *logistic regression* merupakan model yang digunakan mampu mengenali alfabet SIBI dengan akurasi 97% dan 96%. Tetapi, kedua model tersebut masih tetap memiliki kekurangan dalam mengenal alfabet yang gesturnya hampir serupa.

Penggunaan Mediapipe juga dapat membantu menaikkan performa akurasi model yang dibangun dengan metode pretrained ResNet-50[9]. Penelitian ini menggunakan dataset citra 1200 gambar kata isyarat SIBI yang terdiri dari kata

“saya”, “dia”, “kamu”, “cinta”, “sedih”, dan “maaf”. Tujuan dari penelitian ini adalah membandingkan model pengenalan isyarat kata SIBI yang dataset citranya diekstraksi fitur Mediapipe dengan yang tidak. Model yang tidak menggunakan Mediapipe memiliki akurasi sebesar 33% saat *training* dan 40% saat *validation*. Tetapi, model akurasi meningkat menjadi 88% saat *training* dan 87% saat *validation* jika model yang datasetnya diekstrak terlebih dahulu dengan Medipipe sebelum dimasukkan ke dalam tahapan pembuatan model.

Penelitian yang menggunakan ANN untuk pengenalan SIBI juga telah dilakukan pada penelitian[11], kelas yang digunakan pada penelitian ini berjumlah 39 yang terdiri dari 26 alfabet, 10 angka, dan 3 tanda baca. Dataset yang digunakan berupa citra gambar yang jumlahnya sebanyak 3900 dengan resolusi masing-masing citra gambar sebesar 20 x 20 piksel, setiap masing-masing kelas memiliki 100 data citra gambar. Penelitian ini melakukan tahapan normalisasi terhadap dataset yang berupa citra gambar sebelum dimasukkan ke ANN, normalisasi dilakukan dengan mengubah representasi citra gambar menjadi biner. Piksel yang berwarna putih diubah menjadi nol, dan piksel berwarna hitam menjadi satu. Pengujian yang dilakukan menggunakan 10 citra gambar dari setiap kelas yang diambil secara langsung menggunakan *webcam* dari objek yang berjarak 50 cm, hasil yang didapatkan berupa model memiliki performa akurasi sebesar 90% dan menemukan pengaruh pencahayaan saat mengambil citra gambar yang dapat mengurangi performa akurasi model .

Perbedaan penelitian ini dengan yang sebelumnya terletak pada obyek yang digunakan. Penelitian yang telah banyak dikembangkan cenderung menggunakan isyarat abjad BISINDO maupun SIBI, masih sedikit penelitian yang menggunakan kata-kata isyarat yang terdapat di kedua sistem bahasa isyarat tersebut. Pada Tabel 2.1 menunjukkan penelitian-penelitian terdahulu yang berkaitan dengan pengenalan bahasa isyarat dan gestur tubuh.

Table 2.1 Penelitian Terdahulu

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
1	E. Altiarika and W. P. Sari, 2023	Pengembangan Deteksi Realtime untuk Bahasa Isyarat Indonesia dengan Menggunakan Metode *Deep Learning Long Short Term Memory* dan *Convolutional Neural Network*	Sedikitnya penelitian tentang perkembangan teknologi pengenalan bahasa isyarat secara real time untuk memudahkan komunitas tunarungu di Indonesia saling berkomunikasi satu lain menggunakan berbagai bahasa isyarat dan gerak tubuh.	Mencari faktor-faktor yang mempengaruhi tingkat akurasi penerapan objek deteksi dan klasifikasi gambar dan video secara *real time* untuk Bahasa Isyarat Indonesia (BISINDO).	*Long Short Term Memory* dan *Convolutional Neural Network*	Menggunakan data primer yang berupa citra gambar yang berjumlah 1100 dengan ukuran setiap citra sebesar 50 x 50 piksel.	Algoritma CNN yang digabungkan dengan LSTM mampu mengenal isyarat abjad BISINDO dengan akurasi pada saat pelatihan model sebesar 90%, dan pada saat *validation* sebesar 80%. Tetapi, didapatkan juga model memiliki kelamahan hanya bisa mengenal isyarat yang

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
							diperagakan dengan tangan saja, tidak dengan gerakan wajah, mulut, dan tubuh.
2	L. Arisandi et al., 2022	Sistem Klarifikasi Bahasa Isyarat Indonesia (BISINDO) Dengan Menggunakan Algoritma *Convolutional Neural Network*	Terdapat masyarakat yang memiliki keterbatasan fisik yang membuat tidak bisa berkomunikasi secara verbal, oleh karena itu diperlukannya alat untuk mengatasi masalah tersebut.	Merancang sistem klasifikasi BISINDO menggunakan algoritma CNN.	*Convolutional Neural Network*	Dataset memiliki 2163 gambar dan kelas yang terdiri dari abjad BISINDO A-E dan isyarat saya, kamu, dan *I love you*.	Penggunaan algortima CNN dalam merancang sistem klasifikasi BISINDO dapat menghasilkan tingkat akurasi yang sangat tinggi, mencapai 99.82%.
3	Susanty, Meredita Fadillah, Riestiya Zain	Model Penerjemah Bahasa Isyarat Indonesia	Kurangnya aksesibilitas penyandang tuli karena banyaknya	Membangun sistem penerjemah BISINDO	*Convolutional Neural Network* dan	87000 gambar alfabet dalam ASL dan 2659 gambar abjad	Model yang telah dilatih dengan dataset ASL

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
	Irawan, Ade, 2021	(BISINDO) Menggunakan Pendekatan *Transfer Learning*	masyarakat yang belum paham bahasa isyarat untuk bisa saling berkomunikasi.	menggunakan CNN untuk meningkatkan aksesibilitas individu penyandang tuli.	*Transfer Learning*	dalam BISINDO yang merupakan hasil augmentasi.	mendapatkan akurasi 96%. Tetapi,ketika model tersebut diterapkan *transfer learning* terhadap dataset BISINDO, akurasi menjadi 30%. Hal ini dikarenakan perbedaan karakteristik dan gestur antara dua bahasa isyarat, sehingga model hanya bisa mengenali abjad BISINDO

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
							yang mirip dengan ASL.
4	Suyudi, I Sudadio, S Suherman, S, 2023	Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model *Random Forest* dan *Multinomial Logistic Regression*	Jumlah masyarakat umum yang memahami bahasa isyarat yang terbatas, mengakibatkan sedikitnya penutur bahasa isyarat yang membantu penyandang tunarungu berkomunikasi dengan masyarakat umum.	Melakukan pengenalan bahasa isyarat dengan kamera RGB biasa dan kerangka kerja MediaPipe dengan model *random forest* dan *multinomial logistic regression.*	Mediapipe*, Random Forest,* dan *Multinomial Regression*	1000 gambar alfabet SIBI yang memiliki gestur statis dan diperagakan dengan tangan kanan dan kiri.	Model *random forest* dengan parameter jumlah *tree* sebanyak 100 dan model *logistic regression* yang memiliki asimtot pada maksimum iterasi 250 dan L2 regularisasi mampu mengenal abjad SIBI dengan akurasi 97% dan 96%. Tetapi, kedua model memiliki kelemahan

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
							dalam mengenal abjad yang gesturnya hampir serupa.
5	N. Anam, 2022	Sistem Deteksi Simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Mediapipe dan *Resnet*-50	Kesenjangan komunikasi antara penyandang tunarungu atau tunawicara dengan orang normal yang tidak memahami bahasa isyarat yang mengakibatkan salah pemahaman makna dan persepsi diantara keduanya.	Mengetahui performa sistem untuk pengenalan isyarat SIBI yang menggunakan *pre-trained* model *ResNet*-50 dan teknologi Mediapipe.	*ResNet*-50 dan Mediapipe	Citra gambar isyarat SIBI sebanyak 1200 citra yang terdiri dari 6 kata isyarat, yaitu saya, dia, kamu, cinta, maaf, dan sedih.	Performa akurasi yang didapatkan sistem dengan *ResNet*-50 saja sebesar 33% saat *training* dan 40% saat *validation*. Tetapi, model akurasi meningkat menjadi 88% saat *training* dan 87% saat *validation* jika model digabungkan dengan Medipipe.

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
6	Hendapratama, Ikhsanico Hamzah, Ida Wahidah Astuti, Sri, 2022	Rancang Bangun Aplikasi Penerjemah SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Algoritma *Random Forest Classifier*	Masyarakat umum banyak yang sulit memahami bahasa isyarat penyandang tunarungu yang mengakibatkan kesenjangan sosial.	Mengembangkan penerjemahan bahasa isyarat menggunakan *random forest classifier* yang diharapkan dapat membantu masyarakat umum berkomunikasi dengan penyandang tunarungu.	*Random Forest Classifier*	Dataset yang diekstrak menggunakan Mediapipe dari kumpulan citra yang memiliki 10 kelas alfabet BISINDO dari A-J.	Dengan menggunakan algoritma *random forest* mampu mengenal abjad BISINDO dengan akurasi yang tinggi. Tetapi, dibutuhkan peragaan isyarat abjad BISINDO yang presisi saat melakukan pengenalan secara *real time.*
7	S, Nur Budiman *et al*., 2022	Pengenalan Gestur Gerakan Jari Untuk Mengontrol	Masih banyaknya kendala dalam pengenalan *hand gesture* secara *real*	Menerapkan *library* OpenCV dan Mediapipe dalam	Mediapipe	Data primer yang berupa citra berbagai pose tangan	Hasil dari pengujian dari berbagai pose gerakan jari

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
		Volume Di Komputer Menggunakan *Library* Opencv Dan Mediapipe	*time* yang mengakibatkan tingkat akurasi dalam pengenalannya belum optimal.	pengenalan pola gerakan jari tangan manusia untuk mengontrol volume media suara pada perangkat laptop atau komputer secara real time		kanan manusia yang diambil menggunakan *webcam* secara *real time*.	memperoleh akurasi sebesar 88,89%. Tetapi, hasil dari penelitian ini juga terdapat kelemahannya dimana akurasi sangat kecil jika objek tangan terlalu dekat dengan kamera.
8	L. Yusnita, R. Roestam, and R. B. Wahyu, 2017	*Implementation of Real-Time Static Hand Gesture Recognition Using Artificial Neural Network*	Bagaimana implementasi pengenalan gestur tangan statis dalam Sistem Isyarat Bahasa Indonesia (SIBI) dapat memberikan solusi alternatif untuk komunikasi bagi	Menyediakan solusi alternatif bagi komunikasi individu dengan gangguan bicara dan pendengaran menggunakan teknologi pengenalan	*Artificial Neural Network* dan *Feature extraction*	Data primer yang berupa citra gambar yang berjumlah 3900 dengan ukuran setiap citra sebesar 20x20 piksel	Model memperoleh akurasi sebesar 90% dari pengujian menggunakan citra gambar yang diambil menggunakan

No	Penulis, Tahun	Judul	Masalah Penelitian/Rumusan Masalah	Tujuan	Metode	Data	Hasil / Temuan / Kesimpulan
			individu dengan gangguan bicara dan pendengaran.	gestur tangan statis		dan 39 kelas yang terdiri dari 26 alfabet, 10 angka, dan 3 tanda baca.	*webcam* dengan jarak 50 cm dari objek. Tetapi, peerforma model berkurang dikarenakan pencahayaan yang kurang ketika mengambil objek citra gambarnya.

2.2 Landasan Teori

2.2.1 Sistem Isyarat Bahasa Indonesia

Seperti yang sudah dijelaskan pada bagian pendahuluan, penelitian ini memilih Sistem Isyarat Bahasa Indonesia (SIBI). Berdasarkan *website* kamus SIBI yang diterbitkan Kemendikbud, sejarah terbentuknya SIBI didasarkan pada permasalahan yang belum optimalnya penggunaan alat bantu dengar bagi anak tunarungu dan tuli dalam perkembangan penguasaan bahasa dan kemampuan berbicara mereka[20].

Penanganan permasalahan tersebut menggunakan pendekatan komunikasi total yang merupakan pendekatan yang memanfaatkan segala media komunikasi dalam kegiatan pembelajaran tunarungu dan tuli. Dalam pendekatan komunikasi total tidak hanya menggunakan media komunikasi yang sudah lazim seperti berbicara, menulis, membaca, dan mendengar, penggunaan isyarat alami, abjad jari, dan isyarat yang dibakukan juga termasuk dalam pendekatan komunikasi ini.

Pada tahun 1982 dimulai penelitian dan pengembangan perangkat isyarat yang baku dan dapat digunakan secara nasional. Sampai pada tahun 1992 terbentuklah kamus isyarat bahasa Indonesia (ISYANDO), kamus ini tersusun berdasarkan isyarat lokal/alami dari kaum tunarungu di sebelas kota Indonesia yang dikumpulkan dan mengumpulkan isyarat yang diadaptasi dari bahasa isyarat manca negara.

Dalam rangka memperingati hari pendidikan nasional tanggal 2 Mei 1994, Mendikbud yang menjabat pada periode itu mengeluarkan kamus sistem isyarat bahasa Indonesia (SIBI) edisi pertama. Tidak lama setelah kamus SIBI edisi pertama terbit, pemerintah menetapkan kamus tersebut wajib diterapkan oleh seluruh SLB tunarungu yang menerapkan komunikasi total.

SIBI tidak dipelajari secara terpisah, tetapi digunakan bersama dengan metode komunikasi lainnya selama proses pembelajaran. Ini membuat pesan komunikasi menjadi lebih jelas dan bermakna bagi siswa tunarungu. Dengan demikian, diharapkan siswa tunarungu menguasai bahasa Indonesia dengan baik dan benar, memiliki kemampuan literasi, memiliki kesempatan yang lebih besar untuk menyerap kurikulum, lebih mudah mengikuti pendidikan inklusi, dan

akhirnya mampu berintegrasi dalam masyarakat luas. Pada gambar 2.1 menunjukan beberapa gerakan bahasa isyarat SIBI.

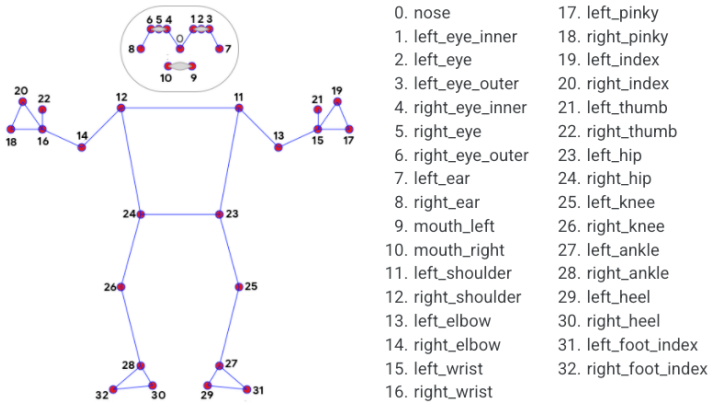
Gambar 2.1 Beberapa Contoh Gerakan Isyarat SIBI

2.2.2 Mediapipe Holistic

Berdasarkan *website* dokumentasinya, Mediapipe merupakan *framework* yang dikembangkan oleh Google untuk menganalisis postur tubuh dan tindakan manusia, *framework* ini dapat memproses data persepsi dari data yang berformat video maupun audio. Dengan memanfaatkan *machine learning, framework* ini dapat melakukan berbagai tugas seperti *hand landmark*, *hand gesture recognition, pose landmark detection* dan masih banyak lagi yang dapat dilihat pada *website*[21].

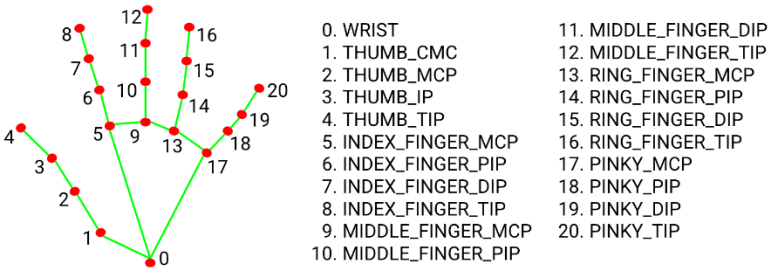
Mediapipe bekerja dengan menangkap gambar *frame by frame* melalui kamera atau *webcam* yang kemudian mendeteksi dan memprediksi pergerakan

kerangka manusia, setelah itu Mediapipe membentuk *pipeline* atau pipa untuk menggambarkan kerangka dalam tubuh manusia yang tertangkap di kamera atau *webcam* yang digunakan. Penggambaran pipa berdasarkan postur tubuh yang terdiri dari 33 titik kunci yang lokasinya diperlihatkan pada gambar 2.2, titik-titik tersebut merupakan titik sendi yang menjadi tumpuan pergerakan tubuh manusia yang diistilahkan dengan *landmark*.



Gambar 2.2 *Landamark* Tubuh Manusia

Selain bisa mendeteksi postur tubuh manusia secara keseluruhan atau *pose landmark detection*, ada salah satu tugas yang bisa dikerjakan Mediapipe juga yaitu pendeteksian pergerakan tangan atau *hand gesture recognition*. Ada 21 *keypoints* pada tangan manusia yang bisa dideteksi dan diprediksi untuk menggambarkan pipa pada pergerakan tangan manusia. Untuk lokasi-lokasi *keypoints* yang dapat dideteksi pada tangan, bisa dilihat pada gambar 2.3 terdapat nomor dan nama yang menjadi *keypoints*.



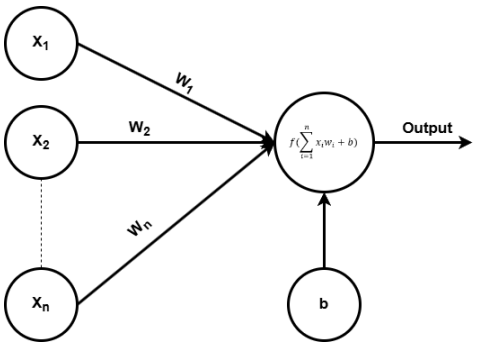
Gambar 2.3 *Landmark* Tangan Manusia

Landmark yang terdeteksi oleh Mediapipe bisa kita ambil informasinya berupa nomor dan koordinat *x, y, z* yang merupakan letak dari *landmark* terhadap gambar yang dideteksinya, koordinat *x, y, z* bisa berubah-rubah jika terjadi

perubahan pada gambar yang sedang dideteksi. Dengan mengumpulkan informasi *landmark* di setiap frame dalam suatu video, bisa menjadi suatu kumpulan data yang bisa dimanfaatkan untuk menyelesaikan masalah seperti dalam pengenalan gerakan manusia pada suatu video.

2.2.3 *Artificial Neural Network*

Algoritma ini terinspirasi dari cara kerja sel otak manusia dalam pembelajaran informasi yang dimiliki untuk mengambil suatu keputusan[22]. Jaringan saraf tiruan atau *artificial neural network* terdiri dari 1 atau lebih neuron yang saling terhubung, sehingga dapat berfungsi untuk memproses data dalam memecahkan kasus seperti pengenalan pola, regresi, dan klasifikasi.



Gambar 2.4 Ilustrasi *Artificial Neural Network*

Neuron merupakan komponen dasar ANN yang menerima *input*, melakukan perhitungan jumlah perkalian antara nilai *input* dengan bobot dan juga pertambahan dengan nilai bias sebagai kontrol, *output* yang keluar diteruskan melalui *activation function*. Operasi perhitungan yang terjadi di dalam neuron ditunjukkan dalam persamaan 2.1.

$$o = f(\sum_{i=1}^{n} x_i w_i + b) \tag{2.1}$$

Notasi:

o : Nilai *output* neuron

f : *activation functions*

o : Nilai *outputs*

x_i : Nilai *input* ke *i*

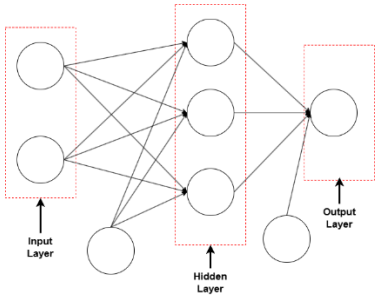
w_i : Nilai bobot ke *i*

b : Nilai bias

n : Jumlah data *input*

2.2.3.1 *Multilayer Perceptron*

Multilayer Perceptron (MLP) merupakan salah satu arsitektur ANN yang memiliki komponen-komponen utama terdiri dari neuron, *layer,* bobot, dan *activation function*. Komponen *layer* atau lapisan yang dimiliki oleh MLP terdiri dari tiga jenis yaitu *input layer, hidden layer,* dan *output layer*. Setiap *layer* bisa tersusun dari satu neuron atau lebih dan saling terhubung dengan *layer* lainnya, ilustrasi arsitektur MLP ditunjukan pada gambar 2.5. Pada *input layer* berfungsi hanya untuk menerima *input* data tanpa melakukan operasi apapun yang kemudian diteruskan ke *hidden layer*.



Gambar 2.5 Ilustrasi Arsitektur MLP

Jenis lapisan selanjutnya yaitu *hidden layer* terjadi proses operasi perhitungan di tiap-tiap neuron terhadap data *input*, hasilnya dilanjutkan lagi ke lapisan berikutnya untuk menjadi *input* lapisan tersebut. *Output* dari *input layer* akan menjadi *input* bagi *hidden layer*, begitupun seterusnya dari *hidden layer* akan mengirimkan hasilnya untuk *output layer*[22].

Lapisan yang terakhir yaitu *output layer* berfungsi untuk menghasilkan prediksi atau keputusan akhir dari model, jumlah neuron di lapisan ini tergantung dengan tujuannya model dibuat. Misalnya model dibuat untuk menyelesaikan masalah klasifikasi, maka jumlah neuron pada *output layer* mengikuti jumlah kelas yang digunakan.

Neuron dalam MLP dapat melakukan pembelajaran untuk menghasilkan *output* terbaik karena dilakukan proses pembelajaran dengan langkah-langkah sebagai berikut[23]:

1. Inisialisasi nilai bobot dan bias dengan nilai *random* atau dengan aturan tertentu.
2. Masukkan nilai *input* ke neuron yang kemudian kita akan mendapatkan nilai *output*, proses ini dinamakan juga dengan *feedforward*.
3. Bandingkan nilai *output* yang didapatkan dengan nilai yang sebenarnya. Jika nilai keduanya sama, maka tidak perlu melakukan perubahan apapun.
4. Apabila nilainya keduanya tidak sama, maka hitung *error* atau *loss* kedua nilai tersebut yang bisa dilakukan dengan persamaan 2.3. Kemudian ubah nilai bobot dan bias menggunakan proses *backpropagation* dengan berdasarkan nilai *error* yang didapatkan.
5. Ulangi langkah-langkah ini sampai tidak ada perubahan pada nilai *error* lagi, nilai *error* kurang dari batas yang telah ditentukan, atau juga sudah melakukan proses pelatihan ini sebanyak nilai yang telah ditentukan.

Salah satu cara menginisialisasi nilai bobot dengan menggunakan nilai dari distribusi *glorot uniform*, ini digunakan untuk mendapatkan nilai bobot yang dapat menjaga variansi *output* di setiap lapisan tetap[24]. Distribusi *glorot uniform* mengambil nilai yang terdapat di rentang [-*limit, limit*] dimana *limit* didefinisikan sebagai persamaan 2.2.

$$\text{limit} = \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}} \tag{2.2}$$

Notasi:

n_{in} : Jumlah unit yang masuk ke neuron

n_{out} : Jumlah unit yang keluar dari neuron

Perhitungan *error* digunakan untuk mengetahui selisih hasil yang dikeluarkan saat proses pembelajaran dengan data sebenarnya, menghitung nilai *error* bisa menggunakan persamaan 2.3.

$$E = y - o \tag{2.3}$$

Notasi:

E : Nilai *error*

y : Data yang sebenarnya

Perhitungan *error* atau *loss* yang menggunakan persamaan 2.3 merupakan cara dasar untuk mengukur selisih hasil prediksi model dengan data aktualnya, seringkali dalam pembuatan model terdapat kasus klasifikasi *mullticlass* seperti pada penelitian ini yang tidak bisa menggunakan persamaan tersebut. Diperlukan metrik yang dapat mengevaluasi model *multicalass,* salah satu metrik yang bisa digunakan yaitu fungsi *loss sparse categorical crossentropy* (SCCE)[25] dengan perhitungannya menggunakan persamaan 2.4.

$$\text{SCCE} = -\sum_{i=1}^{N} y_i \log \hat{y}_i \tag{2.4}$$

Notasi:

N : Jumlah kelas

$\hat{y}$: Prediksi probabilitas dari model untuk setiap kelas

MLP dapat melakukan pembelajaran lebih baik karena menggunakan algoritma *backpropragation*[26]. Pembelajaran yang dimaksud yaitu melakukan perubahan nilai bobot dan bias secara bertahap dari *output layer* ke *hidden layer* berdasarkan nilai *error* atau *loss*. Dengan begitu perubahan nilai bobot suatu *layer* dipengaruhi oleh perubahan nilai bobot dari *layer* setelahnya. Perhitungan untuk mengubah nilai bobot dan bias menggunakan persamaan 2.7 dan 2.8.

$$\Delta w = \alpha \times \delta \times o \tag{2.5}$$

$$\Delta b = \alpha \times \delta \tag{2.6}$$

$$w_{new} = w_{old} + \Delta w \tag{2.7}$$

$$b_{new} = b_{old} + \Delta b \tag{2.8}$$

Notasi:

Δw : Perubahan nilai bobot

Δb : Perubahan nilai bias

α : *learning rate*

δ : Nilai *error* pada neuron

w_{old} : Nilai bobot awal

w_{new} : Nilai bobot baru

b_{old} : Nilai bias awal

b_{new} : Nilai bias baru

Nilai α merupakan *learning rate* yang digunakan untuk menentukan seberapa besar perubahan nilai yang digunakan untuk memperbarui bobot dan bias dalam model, nilai *learning rate* yang terlalu kecil akan membuat waktu pelatihan model semakin lama dan *learnig rate* yang terlalu besar membuat waktu pelatihan model lebih cepat tetapi saat memperbarui nilai bobot dan bias menjadi kurang optimal[27].

Nilai δ merupakan *error* yang dimiliki oleh neuron setiap setelah melakukan proses *feedforwad*. Ada perbedaan cara mencari nilai δ pada neuron yang berada di *output layer* dan di *hidden layer*, pada *output layer* mencari nilai δ menggunakan persamaan 2.9 dan pada *hidden layer* menggunakan persamaan 2.10.

$$\delta_o = o \times (1 - o) \times E \tag{2.9}$$

$$\delta_h = o \times (1 - o) \times (\delta_o \times w) \tag{2.10}$$

Notasi:

δ_o : Nilai *error* neuron pada *output layer*

δ_h : Nilai *error* neuron pada *hidden layer*

Komponen *activation function* berguna untuk menentukan suatu neuron akan aktif atau tidak dan juga membuat ANN dapat memodelkan hubungan kompleks dalam data, hal ini dikarenakan *activation function* memperkenalkan unsur *non-linearitas* ke dalam ANN[28]. Ada beberapa macam *activation function* yang sering digunakan seperti *sigmoid* yang mengeluarkan nilai yang memiliki rentang dari 0 sampai 1, fungsi ini sering digunakan untuk *output* yang dapat diinterpretasikan dengan probabilitas seperti menyelesaikan permasalahan klasifikasi. *Sigmoid function* memliki persamaan matematika yang dituliskan pada persamaan 2.11.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.11}$$

Notasi:

σ : Nilai *output* probabilitas

e : Bilangan konstan *Euler*

x : Nilai *input* yang masuk ke fungsi *sigmoid*

Fungsi selanjutnya ada *rectified linear unit* (ReLU) yang menghasilkan keluaran yang sama dengan nilai *input* jika nilai *input* tersebut positif, dan 0 jika

nilai *input* negatif. Fungsi lainnya lagi yaitu *softmax* yang menghasilkan keluaran probabilitas setiap kelasnya[25], fungsi ini sering digunakan pada kasus klasifikasi *multiclass* dan memliki persamaan matematika yang dituliskan pada persamaan 2.12.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{J=1}^{K} e^{z_J}} \tag{2.12}$$

Notasi:

$\sigma(z)_i$: Nilai *output* probabilitas setiap kelas *i*

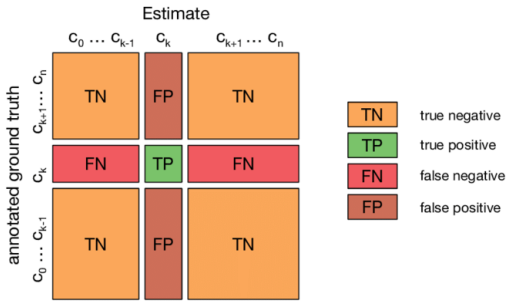
z : Vektor *input* ke fungsi *softmax*

z_i : Elemen ke *i* dari vektor *input z*

K : Jumlah kelas

2.2.4 *Confusion Matrix Multiclass*

Ini adalah metode untuk menampilkan dan membandingkan nilai sebenarnya dengan nilai hasil prediksi suatu model dan menghasilkan matrik evaluasi seperti *accuracy*, penggunaan matrik ini akan memberitahukan informasi seberapa baik model yang telah dibuat[29]. Selain digunakan untuk model klasifikasi dua kelas, *confusion matrix* juga bisa digunakan untuk model klasfikasi dengan jumlah kelas lebih dari dua[30].



Gambar 2.6 Tabel *Confusion Matrix Multiclass*

Berdasarkan gambar 2.6, nilai matrik *accuracy* bisa dihitung menggunakan persamaan 2.12. *Accuracy* didapatkan dengan membandingkan jumlah data yang diprediksi benar dengan total seluruh prediksi yang dilakukan, matrik ini menggambarkan seberapa sering model membuat prediksi yang benar.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.13)$$

Notasi:

C_n : Kelas ke-n

TP (*True Positif*) :Jumlah data diprediksi kelas C_n dan memang sebenarnya kelas C_n.

TN (*True Negative*) :Jumlah data diprediksi bukan kelas C_n dan memang sebenarnya memang bukan kelas C_n.

FP (*False Positif*) :Jumlah data diprediksi kelas C_n tetapi sebenarnya data bukan bernilai kelas C_n..

FP (*False Negative*) :Jumlah data diprediksi bukan kelas C_n tapi sebenarnya data tersebut bernilai kelas C_n.

2.2.5 *Overfit* dan *Underfit*

Overfit merupakan kondisi model yang telah dilatih mampu prediksi data *training* sangat bagus, namun tidak mampu memprediksi secara akurat pada data *testing* atau data uji. Model yang *overfit* tidak dapat mengeneralisasi data yang baru ditemui sehingga hasil prediksi yang didapatkan banyak yang salah. Sebaliknya, *underfit* terjadi ketika model terlalu sederhana untuk menangkap pola yang ada dalam data *training,* sehingga tidak mampu menghasilkan prediksi yang akurat saat menggunakan data *training* maupun *testing*[31].