BAB 2
LANDASAN TEORI

Pada bagian ini terdapat beberapa teori yang akan dibahas dalam penelitian kali ini. Pada penelitian kali ini materi kemampuan berbicara ke teks akan dibahas. Berikut akan ada teori-teorinya pada bagian di bawah ini.

2.1 Berbicara

Berbicara adalah suatu hal yang dilakukan oleh manusia dalam kehidupan sehari-hari. Berbicara merupakan keterampilan untuk berkomunikasi [17], bertukar informasi [10], dan mengenali orang lain pada waktu yang sebenarnya. Berbicara perlu dilatih oleh manusia sejak dini. Latihannya berbicara kali ini berupa terapi wicara, terapi perilaku, dan menerapkan pola intervensi pada anak usia dini [7]. Latihan tersebut dilakukan setiap hari untuk meningkatkan keterampilan berkomunikasi dengan menyediakan transkripsi ucapan pada saat yang sebenarnya [17].

Latihan berbicara akan dilakukan menggunakan aplikasi PyCharm. Aplikasi ini digunakan untuk melatih kemampuan berbicara manusia selama beberapa periode dengan jumlah satu sampai dua kata. Latihan berbicara dilakukan dengan menggunakan bahasa pemrograman Python. Latihan tersebut dilakukan untuk membuat himpunan data suara dengan lebih dari 10 kata yang berbeda. Setiap kata yang berbeda akan dilakukan pengujian data dengan sampel kata yang sama sebanyak minimal 100 sampel. Sampel yang diambil diharapkan memperoleh hasil yang sesuai dan tingkat akurasi yang tinggi terhadap data yang dibuat.

Sampel ucapan suara seseorang disimpan pada penyimpanan berupa audio. Audio tersebut akan mengenali ucapan suara seseorang secara andal. Audio tersebut akan dilakukan transkripsi dan disimpan pada sebuah berkas [17]. Berkas tersebut merupakan hasil pembicaraan yang dilakukan oleh seseorang menggunakan bahasa pemrograman Python dan akan disimpan dalam format WAV.

2.2 Wicara

Wicara adalah pembicaraan yang berasal dari mulut manusia. Wicara adalah suatu hal yang sangat penting dalam kehidupan sehari-hari. Wicara sangat diperlukan terutama saat melakukan interaksi, menyampaikan pesan, dan bertukar

informasi. Untuk itu, suara seseorang perlu dikenal secara mendalam melalui metode pembelajaran mendalam dengan algoritma *Convolutional Neural Network* (CNN) [18].

Pengenalan suara pada konteks kali ini menggunakan aplikasi PyCharm dan bantuan mikrofon. Seseorang akan membuat kode program terlebih dahulu. Kode program yang dibuat akan dikukuhkan melalui terminal. Setelah kode program tersebut dikukuhkan, aplikasi tersebut akan dijalankan.

Kronologis aplikasi yang dijalankan akan dilakukan dengan cara seseorang menunggu kesempatan untuk berbicara pada terminal. Jika kesempatan berbicara sudah muncul, maka seseorang dapat berbicara. Waktu berbicara yang dapat dilakukan oleh seseorang maksimal dua menit tanpa berhenti. Jika seseorang mau berhenti berbicara, seseorang dapat berhenti selama maksimal 0,8 detik. Jika seseorang berhenti berbicara selama lebih dari 0,8 detik, maka program tersebut akan menangkap semua suara yang telah dibicarakan oleh seseorang [14]. Semua pembicaraan yang telah diucapkan oleh seseorang akan disimpan melalui berkas audio. Berkas audio tersebut adalah WAV [19].

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah sebuah algoritma model yang digunakan untuk memperkirakan emosi yang tertanam pada sinyal ucapan seseorang [20]. CNN berhubungan dengan gambar [21], pengujian data, pengujian spektogram, pengujian MFCC, dan pengenalan suara. Hal tersebut bisa diterapkan dalam pengenalan aktivitas manusia mengenai penerjemahan suara manusia ke teks [22].

Algoritma CNN juga memiliki arsitektur. Ada empat lapisan utama dalam arsitektur CNN. Lapisan tersebut ialah lapisan konvolusi, lapisan *pooling*, lapisan aktif, dan lapisan terhubung penuh [1, 23, 24]. Keempat lapisan tersebut memiliki maksud dan fungsi yang berbeda. Penjelasan tersebut dapat dilihat pada bagian di bawah ini.

1. Lapisan Konvolusi

   Lapisan ini merupakan operasi pencarian fitur yang digunakan pada gambar masukan melalui proses filtrasi. Pada lapisan ini, gambar dapat diperoleh dengan menggerakkan filter kecil seperti filter dua kali dua, empat kali empat, atau lima kali lima menggunakan kernel [1, 23]. Representasi dari lapisan konvolusi dapat dilihat pada Gambar 5.1 pada halaman lampiran.

2. Lapisan *Pooling*

   Lapisan *Pooling* adalah lapisan yang digunakan untuk mengurangi ukuran gambar. Ukuran gambar dikurangi dengan mengambil nilai terbaik dari gambar tersebut, demi mengurangi beban komputasi dan mencegah kelebihan muatan pada sistem [1, 23]. Representasi dari lapisan *pooling* dapat dilihat pada Gambar 5.2 pada halaman lampiran.

3. Lapisan Aktif

   Lapisan ini bertujuan untuk menyelesaikan masalah yang lebih kompleks dari jaringan saraf. Lapisan ini memiliki empat fungsi aktivasi, yaitu:

   - Fungsi sigmoid: fungsi matematis yang digunakan untuk jaringan saraf yang sangat tua yang menghasilkan rentang keluaran antara nol sampai satu. Fungsi sigmoid dapat diperoleh dengan Formula 2.1.

     $$sigma(x) = 1/(1+e^{-x}) \tag{2.1}$$

     Pada Formula 2.1 [1], $sigma(x)$ adalah fungsi sigmoid dari variabel x, x adalah masukan dari fungsi sigmoid, dan e adalah bilangan Euler (e kira-kira 2.71828) yang merupakan bilangan konstan dalam Matematika [1]. Fungsi sigmoid dapat dilihat pada Gambar 5.3 bagian kiri atas pada halaman lampiran.

   - Fungsi Tanh (*Hyberbolic Tangent Function*): fungsi matematis yang digunakan untuk mengatasi masalah gradien yang hilang pada sigmoid. Hasil keluaran dari fungsi ini berada pada kisaran negatif satu sampai plus satu. Fungsi sigmoid dapat diperoleh dengan Formula 2.2.

     $$Tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \tag{2.2}$$

     Pada Formula 2.2 [1], Tanh(x) adalah hasil dari fungsi sasaran hiperbolis pada nilai x. e^{x} merupakan fungsi eksponensial dari x, dan e^{-x} merupakan fungsi eksponensial dari -x [1]. Fungsi Tanh dapat dilihat pada Gambar 5.3 bagian kanan atas pada halaman lampiran.

   - Fungsi ReLU (*Rectified Linear Unit*): fungsi aktivasi yang cukup populer karena fungsi ini bisa menggantikan nilai negatif menjadi nol dan mempertahankan nilai positif. Fungsi aktivasi diperoleh dengan

Formula 2.3.

$$f(x) = max(0,x) \tag{2.3}$$

Pada Formula 2.3 [1], fungsi ReLU tidak mungkin negatif. Fungsi ReLU negatif jika nilai x negatif atau nol. Fungsi ReLU akan sesuai dengan nilai x jika nilai x positif [1]. Fungsi ReLU dapat dilihat pada Gambar 5.3 bagian kiri bawah pada halaman lampiran.

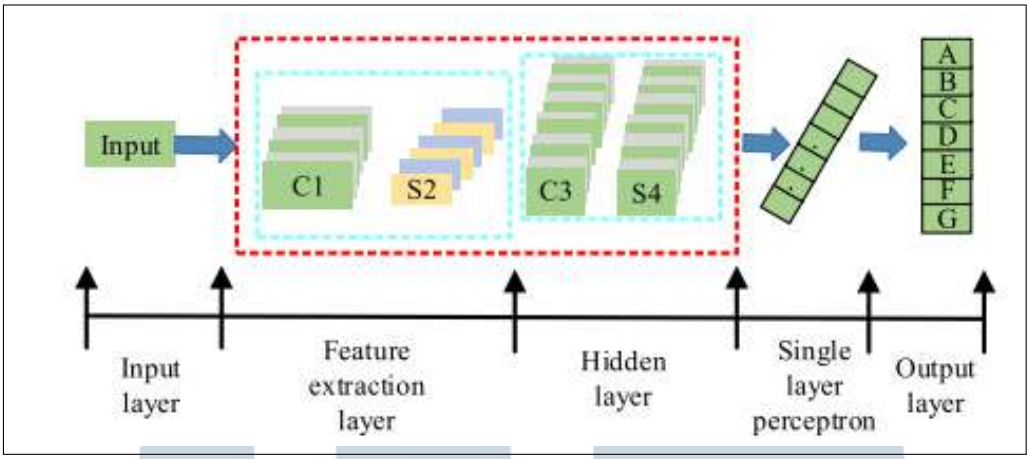
- Fungsi ReLU yang bocor: fungsi untuk mengatasi masalah neuron yang mati dengan membiarkan nilai negatif dapat keterlibatan dengan gradien yang kecil. Fungsi ini diperoleh dengan Formula 2.4.

$$f(x) = max(\alpha(x),x) \tag{2.4}$$

Pada Formula 2.4 [1], fungsi ReLU ini akan mengembalikan nilai yang paling maksimal antara α(x) atau x [1]. Fungsi ReLU dapat dilihat pada Gambar 5.3 bagian kanan bawah pada halaman lampiran.

4. Lapisan Terhubung Penuh
   Lapisan Terhubung Penuh digunakan untuk menggabungkan semua matriks konvolusi menjadi satu kesatuan dengan ukuran satu kali satu [1, 23].

Selain empat lapisan pada arsitektur CNN, ada dua kombinasi algoritma pada arsitektur CNN. Algoritma tersebut ialah algoritma pembelajaran mendalam dan algoritma *Artificial Neural Network* (ANN) yang biasanya dilakukan dalam proses pengelolaan gambar. *Artificial Neural Network* (ANN) adalah lapisan yang terhubung satu sama lain dengan arsitektur CNN. Lapisan tersebut umumnya terdiri dari tiga bagian, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Lapisan masukan adalah lapisan yang asli sebelum gambar diproses. Lapisan tersembunyi adalah lapisan neuron berlapis-lapis dengan struktur yang kompleks dan bukan linear termasuk lapisan konvolusi dan lapisan pengambilan sampel. Sedangkan lapisan keluaran adalah hasil dari klasifikasi gambar dan lapisan tersembunyi [1]. Hasil dari lapisan tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1. Struktur dari lapisan *Convolutional Neural Network* (CNN). Sumber: [1]

Pada Gambar 2.1 terdapat lima lapisan pada strukturnya. Lapisan tersebut dimulai dari lapisan masukan, lapisan fitur ekstraksi yang merupakan lapisan untuk memasukkan pra-proses data, lapisan tersembunyi, lapisan perceptron tunggal yang merupakan lapisan untuk memproses data pada lapisan tersembunyi, dan lapisan keluaran. Kelima lapisan tersebut akan dilakukan konversi menjadi sebuah matriks satu dimensi dengan lapisan yang terhubung penuh [1].

Algoritma *Convolutional Neural Network* (CNN) juga digunakan dalam proses pengujian model. Proses pengujian model pada konteks ini dilakukan dengan pembuatan model menggunakan algoritma CNN. Pembuatan model tersebut dilakukan secara bertahap. Tahapan tersebut dimulai dari model pra-pelatihan. Model pra-pelatihan adalah model untuk mengukuhkan pelatihan dan mencapai hasil yang lebih baik dan optimal [12]. Proses pra-pelatihan tersebut akan dilakukan dengan cara mengambil himpunan data dari internet. Himpunan data tersebut merupakan himpunan data yang berupa terjemahan hasil wicara ke teks.

Tahapan dalam pengujian data Kaggle ada banyak. Tahapan tersebut dimulai dari membuat dan mengambil himpunan data yang diperlukan. Setelah himpunan data diambil, proses mengunduh pustaka seperti data masukan Kaggle, dan sistem operasi diperlukan. Kemudian, proses inisiasi himpunan data dilakukan. Himpunan data yang telah dilakukan inisiasi akan dibagi dalam beberapa sampel. Setelah itu, semua sampel data akan dibagi menjadi himpunan data dalam persen. Lalu, jumlah sampel semua data akan dihitung. Setelah itu, tahapan ekstraksi dan pengujian data dan akurasi akan dilakukan. Proses tersebut terdiri dari pengujian spektogram, pengujian MFCC, pengujian *zero crossing* data, dan pembuatan model [2].

2.3.1 Spektogram

Tantangan terbesar dalam menggunakan algoritma *Convolutional Neural Network* (CNN) adalah dimensi sinyal ucapan. Menanggapi tantangan tersebut, fitur spektral-temporal tiga dimensi harus dipelajari untuk mengubah sinyal audio menjadi satu dimensi. Proses tersebut adalah representasi spektogram yang merupakan proses representasi visual dari kekuatan sinyal dari waktu ke waktu pada frekuensi yang berbeda [20] dengan mengubah sinyal suara dari domain waktu ke domain frekuensi [15].

Pada penelitian sinyal suara, spektogram adalah proses representasi visual dari *Short Time Fourier Transform* (STFT) di mana sumbu horizontal mewakili waktu dan sumbu vertikal mewakili frekuensi pada bingkai yang pendek. Pada titik tertentu, terdapat bagian yang berwarna gelap merupakan data suara yang memiliki frekuensi lebih rendah dan warna terang merupakan data suara yang memiliki frekuensi lebih tinggi. Spektogram ini sangat cocok untuk berbagai analisis, termasuk terjemahan hasil wicara menjadi teks [20].

Pada pengujian spektogram, sinyal suara juga akan diukur. Sinyal suara tersebut akan menggunakan sensor utama yaitu mikrofon. Mikrofon tersebut akan dijadikan alat perekam yang memiliki beberapa ciri-ciri penting, yaitu:

- Frekuensi istirahat: kisaran perubahan amplitudo yang berhubungan dengan frekuensi (tertinggi hingga terendah) dalam Hz yang turut memengaruhi jumlah desibel.
- Rasio sinyal terhadap kebisingan: rasio antara latar belakang suara gangguan yang terekam dengan sinyal yang masuk ke alat perekam.
- Rentang dinamis: kisaran variasi amplitudo pada setiap alat perekam.
- Kecepatan pita: kecepatan rekaman yang menunjukkan kualitas optimum hasil rekaman.

Keempat ciri-ciri ini digunakan agar diperoleh hasil rekaman spektogram yang lebih optimal dengan menggunakan algoritma *Fast Fourier Transform* (FFT) dengan menggunakan algoritma *Discrete Fourier Transform* (DFT). Algoritma ini digunakan untuk melakukan analisis frekuensi gelombang suara. Frekuensi gelombang suara tersebut dapat dilihat pada Gambar 5.4 pada halaman lampiran [3].

2.3.2 Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) merupakan sistem pengenalan suara otomatis yang berkaitan dengan sistem klasifikasi emosi ucapan seseorang. Sistem klasifikasi tersebut terdiri dari klasifikasi fitur dan ekstraksi fitur. Klasifikasi fitur adalah fitur yang berkaitan dengan proses ekstraksi untuk melakukan prediksi kategori emosi. Ekstraksi fitur adalah ekstraksi fitur ucapan seseorang yang berkaitan dengan emosi yang menggunakan beberapa alat ekstraktor. Kedua fitur tersebut telah dilaporkan dan memiliki tiga ciri khas berdasarkan cara manusia berbicara yaitu ciri prosodi, fitur suku kata, dan fitur spektral [25].

Pada *Mel Frequency Cepstral Coefficients* (MFCC), proses klasifikasi emosi ucapan seseorang disarankan menggunakan *Low-Frequency Power Coefficients* (LFPC) untuk mewakili sinyal ucapan dan *Hidden Markov Model* (HMM) untuk melakukan proses klasifikasi. Metode *Hidden Markov Model* (HMM) terkadang kurang efektif karena sering menyebabkan kesamaan antar kata jika jumlahnya banyak [15]. Namun, metode ini tetap digunakan untuk membedakan nada suara dan kecepatan berbicara antara usia anak-anak dan dewasa [25], suara spesies, dan suara hewan [2]. Bentuk dari LFPC dan HMM akan dilakukan untuk memperoleh spektrum dan membagi sinyal suara menjadi beberapa bagian [2], serta memperoleh hasil dari sinyal ucapan seseorang secara mendalam menggunakan metode *Hidden Markov Model* (HMM) [25, 26].

Proses ekstraksi dalam pengujian MFCC memerlukan daya dan energi. Daya dan energi tersebut diolah dengan menggunakan tiga teknik, yaitu:

- Teknik Arsitektur:
  Proses sinyal campuran yang dilakukan mencapai efisiensi dan kecepatan yang lebih cepat dari yang canggih [4].

- Teknik Algoritma:
  Proses realisasi MFCC konvensional diganti dengan proses realisasi yang diusulkan yaitu *Fast Fourier Transform* (FFT) [4].

- Verifikasi Silikon:
  Melakukan operasi filter yang memiliki lintasan tinggi untuk menghemat area dan operasi bingkai untuk merancang realisasi sinyal campuran [4].

Proses ekstraksi dalam MFCC secara umum ditujukan pada Gambar 5.5 pada halaman lampiran. MFCC ini merupakan fitur yang digunakan untuk

melakukan deskripsi energi sinyal dalam domain frekuensi Mel. Ada empat tahapan deskripsi dalam ekstraksi MFCC, yaitu:

- Frontend dan Konversi Data: Pada umumnya, proses untuk melakukan tahapan ekstraksi ini diperlukan domain digital. Untuk itu, proses konversi sinyal ucapan yang masih analog, perlu dilakukan konversi menjadi digital. Pada proses konversi ini, sinyal masukan kontinu v(t) diambil sampel dan dilakukan kuantisasi menjadi sinyal diskrit [4].

- Penekanan awal dan modul penyusunan: Proses untuk melakukan meratakan amplitudo frekuensi tinggi dan amplitudo frekuensi rendah yang disebabkan oleh efek bibir suara yang kabur. Proses ini dapat dilihat pada Gambar 5.6 pada halaman lampiran, di mana pada bagian kiri gambar merupakan frekuensi sinyal MFCC sebelum dilakukan penekanan awal, dan pada bagian kanan gambar merupakan frekuensi sinyal MFCC setelah proses penekanan awal [4].

- Frekuensi Domain Transformasi: Proses mengubah frekuensi sinyal suara dari domain waktu menjadi domain frekuensi [3, 4]. Proses mengubah frekuensi sinyal suara tersebut dapat dilihat pada Formula 2.5.

$$x[k] = |\mathrm{FFT}(x[n])|^2 \tag{2.5}$$

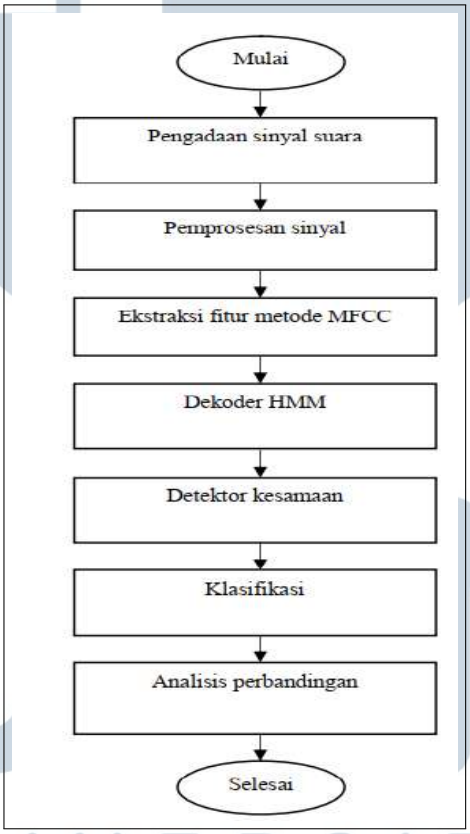
  Pada Formula 2.5 [4], menjelaskan bahwa x[n] adalah sinyal diskrit dalam domain waktu. FFT(x[n]) adalah transformasi Fourier dari sinyal x[n] yang menghasilkan spektrum frekuensi dari sinyal MFCC, x[k] adalah nilai magnitudo atau daya spektrum frekuensi pada frekuensi ke-k. Fungsi pada Formula 2.5 dibuat kuadrat untuk mengukur atau intensitas pada frekuensi tertentu.

- Filtrasi Mel dan Proses Akhir: spektrum filter dari MFCC akan ditunjukkan pada Gambar 5.7 pada halaman lampiran. Pada proses ini pita mel akan melebar jika sinyal MFCC semakin besar, dan pita mel akan menipis jika sinyal MFCC semakin kecil. Lebar pita mel dapat diukur dengan menggunakan Formula 2.6.

$$C[m] = \sum_{k=1}^{k} \cdot \log(X[k]) \cos\left(\frac{\pi mk - 0.5}{K}\right) \tag{2.6}$$

Pada Formula 2.6 [4], log(x[k]) adalah fungsi logaritma berbasis 10 dari x[k], C[m] adalah nilai yang dihitung untuk m. Cos merupakan fungsi kosinus. Sedangkan k dan m adalah parameter yang digunakan pada proses filtrasi ini.

Setelah dilakukan proses ekstraksi, dilakukan tahapan pengujian MFCC. Ada beberapa tahapan dalam pengujian MFCC. Tahapan tersebut terdiri dari pengadaan sinyal suara, proses sinyal, fitur ekstraksi MFCC, dekoder, klasifikasi, dan analisis [2]. Pengujian tersebut dilakukan untuk mendapatkan proses mentah data ucapan dan pengenalan fitur untuk mendapatkan hasil yang efisien [2]. Skenario pengujian dapat dilihat pada Gambar 2.2.



Gambar 2.2. Menguji data dengan MFCC. Sumber: [2]

Berdasarkan Gambar 2.2, hasil pengujian MFCC (*Mel Frequency Cepstral Coefficients*) akan terlihat. Pengujian MFCC dilakukan untuk menghasilkan pengadaan sinyal suara. Sinyal suara tersebut akan dilakukan proses sinyal. Sinyal yang diproses akan dilakukan fitur ekstraksi. Fitur ekstraksi tersebut akan dilakukan menggunakan MFCC. MFCC tersebut akan menghasilkan frekuensi berdasarkan bingkai. Semakin tebal bingkainya, semakin baik frekuensi MFCC. Setelah itu, ada proses pengenalan suara seseorang melalui dekoder. Dekoder digunakan untuk

menghasilkan keluaran sinyal suara yang lebih optimal. Kemudian, ada proses klasifikasi. Proses klasifikasi ini digunakan untuk melakukan identifikasi suara berdasarkan sinyal. Lalu, proses analisis dilakukan untuk mendapatkan akurasi yang lebih tinggi [2].

2.3.3 Zero crossing data

Zero crossing data adalah proses pengujian himpunan data pada fungsi yang mendekati titik nol. Fungsi tersebut merupakan fungsi yang berasal dari pengujian MFCC. Pengujian MFCC tersebut akan dianalisis. Analisis tersebut bertujuan untuk mendapatkan akurasi yang baik dan tingkat kesalahan yang rendah [2]. Tentunya, amplitudo suara akan tampil pada tahapan ini. Semakin tinggi amplitudo, semakin bagus sinyal suaranya.

2.3.4 Pembuatan Model

Pembuatan model adalah proses yang dilakukan dalam pengujian himpunan data. Proses ini dilakukan setelah proses pengujian MFCC. Pengujian ini digunakan untuk melakukan pengujian data yang memiliki jumlah sampel besar. Pengujian data ini dilakukan dalam banyak periode. Semakin banyak periode data yang diuji, maka semakin tinggi tingkat akurasi data yang diperoleh.

Proses pengujian model memerlukan waktu yang lama. Hal ini tergantung dengan banyaknya periode data yang diuji. Pengujian tersebut berupa x_*test*, y_*test*, x_*train*, dan y_*train* [27]. Hasil tersebut akan memunculkan tingkat akurasi berdasarkan nilai positif asli, nilai positif palsu, nilai negatif asli, dan nilai negatif palsu [18, 22, 28]. Tingkat akurasi tersebut akan diukur dengan rumus jumlah data yang secara prediksi benar/ jumlah data seluruhnya) dikalikan 100% [29]. Jumlah data yang benar akan dinilai kebenaran berdasarkan tampilan tingkat akurasi data pada bagian keluaran. Semua hasil keluaran model tersebut akan disimpan sebagai hasil dari pengujian data melalui Kaggle.

2.4 Tensorflow

Pembelajaran mendalam adalah bidang kecerdasan buatan. Pembelajaran mendalam tersebut dibangun dalam proses menggambarkan grafik yang sederhana dengan banyak lapisan [24]. Grafik yang dibuat tersebut akan digunakan dalam melatih *Artificial Neural Network* (ANN) pada berbagai lapisan himpunan data

[24]. Himpunan data tersebut akan diimplementasikan pada pembuatan model melalui aplikasi tensorflow.

Tensorflow merupakan sumber pustaka terbuka yang digunakan untuk perhitungan numerik yang dikembangkan oleh Tim Otak Google. Tensorflow merupakan metode pembelajaran mesin canggih yang menggunakan jaringan saraf dalam [30]. Tensorflow juga memiliki arsitektur fleksibel yang memungkinkan kemudahan implementasi dalam arsitektur yang berbeda (*Central Processing Unit* (CPU), *Graphics Processing Unit* (GPU), dan *Tensor Processing Unit* (TPU)) pada desktop, gugusan, dan perangkat seluler. Pada arsitektur tersebut juga akan ditampilkan hasil dari penggunaan daya CPU dan GPU setelah dilakukan pengujian [24]. Pada penelitian pengujian pesan suara ke teks ini, perangkat tensorflow yang digunakan hanya CPU. Perangkat GPU dan TPU tidak digunakan. Pada perangkat CPU, penggunaan daya CPU dan suhu CPU juga akan ditampilkan pada pengujian model [24].

Tensorflow juga merupakan perpustakaan pembelajaran mendalam yang diimplementasikan berdasarkan arahan grafik. Arahan grafik tersebut mewakili matematika, operasi, dan tepian yang mewakili aliran data antara node yang membuat tensorflow digunakan pada domain mana pun yang dapat dirancang sebagai jaringan aliran perhitungan. Aplikasi Tensorflow diunduh pada aplikasi komputer melalui Windows, Linux, Mac OS, dan pada platform lain seperti Android OS, dan Raspberry [24].

Tensorflow yang akan digunakan pada penelitian kali ini adalah Tensorflow 2.12.0. Tensorflow 2.12.0 merupakan aplikasi untuk membuat model pembelajaran mesin dan model algoritma pembelajaran mendalam. Selain itu, Tensorflow 2.12.0 juga berfungsi untuk analisis data, mengolah data, dan pengenalan penyampaian suara dari seseorang. Pada konteks kali ini, pengenalan penyampaian suara dari seseorang dilakukan dengan cara seseorang berbicara satu sampai dua kata. Kata tersebut akan diterjemahkan ke teks.

Mekanisme penggunaan tensorflow dilakukan dengan cara implementasi perintah suara. Mekanisme tersebut berawal dari memasukkan perintah audio, mengubah audio menjadi teks, dan pengecekan teks dengan perintah yang tersedia. Pada mekanisme pengecekan teks dengan perintah tersebut, perintah suara akan dilakukan pengujian dengan menggunakan *Convolutional Neural Network* (CNN). Pengujian CNN pada tensorflow akan dilakukan secara berlapis. Lapisan yang digunakan adalah lapisan dua dimensi. Lapisan yang diuji pada tensorflow bertujuan untuk mendapatkan sinyal suara yang tepat, akurat, dan mengurangi

kerugian [31].

Pengujian tensorflow ini juga dilakukan melalui aplikasi Visual Studio Code. Pada pengujian ini, matriks konvolusi digunakan untuk menjalankan tensorflow. Matriks konvolusi yang digunakan adalah Python 3.11.3. Selain itu, diperlukan perangkat tambahan dalam pengujian tensorflow. Perangkat tersebut berupa *Compute Unified Device Architecture* (CUDA) dan *CUDA Deep Neural Network* (CuDNN). Kedua perangkat tersebut digunakan untuk mempermudah program tensorflow berjalan pada server CPU [32].

Pengujian tensorflow dilakukan pada proses klasifikasi grafik. Proses tersebut dimulai dari memasukkan node data, dan ukuran atribut yang dimasukkan. Setelah itu, ukuran matriks konvolusi dan grafik konvolusi akan dimasukkan. Grafik konvolusi yang dimasukkan akan dilakukan pengujian aktivasi fungsi matematis yang berupa ReLU(*Rectified Linear Unit*). ReLU bertujuan untuk mendapatkan data yang berkaitan dengan *Artificial Neural Network* (ANN). Setelah semua tahapan tersebut berakhir, pengujian model pada data tersebut akan dilakukan dengan menggunakan kategori Adam secara acak dengan menggunakan perangkat CUDA dan CuDNN. Hasil tersebut akan menampilkan penggunaan daya CPU dan suhu CPU pada tensorflow [24].

2.5 Audio

Audio merupakan fitur yang digunakan untuk pengenalan suara yang sangat diandalkan. Audio merupakan perangkat keluaran yang akan bersuara ketika seseorang telah berbicara. Audio tersebut akan bersuara ketika seseorang telah berbicara. Audio akan mengeluarkan suara diputar. Audio yang diputar ialah audio yang dilakukan proses transkripsi melalui penerjemahan pesan suara secara otomatis [17]. Tujuannya agar manusia dapat dengan mudah mengenali suara seseorang [33].

Audio juga digunakan sebagai sampel dalam pengenalan suara. Audio tersebut bisa mengenali suara seseorang dalam bahasa apa pun, termasuk bahasa Indonesia. Misalnya jika seseorang mengatakan **Budi suka makan cokelat**, maka kata yang akan diucapkan melalui audio adalah **Budi suka makan cokelat**. Jika tidak, maka audio tersebut error atau lafal seseorang dalam berbicara bisa jadi kurang jelas.

2.6 One Hot Encoder

One Hot Encoder merupakan pembelajaran mesin yang digunakan untuk mendapatkan fitur numerik. *One Hot Encoder* bertujuan untuk mendapatkan model yang terbaik berdasarkan hasil pemisahan, pengujian, pelatihan, dan evaluasi model [28].

Pada metode *One Hot Encoder*, proses pelatihan bisa dilakukan memisahkan data untuk dilakukan pengujian yaitu 10%, 30%, atau 60%. Sisa data dari proses pengujian akan digunakan untuk proses pelatihan. Hasil dari pelatihan dan pengujian tersebut akan menghasilkan model evaluasi berupa laporan klasifikasi dan *confusion* matriks [28].

Pada proses pelatihan data, proses optimalisasi akan dilakukan. Proses optimalisasi ini menggunakan kategori Adam dengan kerugian berupa *categorical cross entropy*. *Categorical cross entropy* digunakan untuk menghindari adanya kesamaan dalam label sebenarnya dan model prediksi. Selain itu, *categorical cross entropy* juga bertujuan untuk mendapatkan nilai akurasi yang semakin tinggi dan kerugian yang semakin kecil pada saat melakukan proses kompilasi model. Proses kompilasi ini memerlukan waktu beberapa saat. Setelah menunggu beberapa saat, matriks akurasi akan dihasilkan yang terdiri dari akurasi data dan tingkat kerugian [34].

Pada model evaluasi, ada dua komponen utama yang diperoleh. Komponen tersebut secara umum berupa *confusion* matriks dan laporan klasifikasi.

Pada *confusion* matriks, model evaluasi akan menghasilkan model yang berukuran persegi. Model evaluasi tersebut akan membentuk tabel dengan sumbu vertikal berupa label prediksi dan sumbu horizontal berupa label sebenarnya [28]. Model evaluasi tersebut bertujuan untuk memperoleh tingkat akurasi tinggi dan data hilang rendah berdasarkan nilai positif asli (TP), nilai positif palsu (FP), nilai negatif asli (TN), dan nilai negatif palsu (FN) [2, 28]. Nilai positif asli adalah kondisi sistem bisa mendeteksi ada keluaran pada data tersebut. Nilai negatif asli adalah kondisi di mana sistem tidak dapat mendeteksi ketika tidak ada keluaran pada data tersebut. Nilai positif palsu adalah kondisi di mana sistem bisa mendeteksi tidak ada keluaran yang ada pada data tersebut. Sedangkan, nilai negatif palsu adalah kondisi di mana sistem tidak bisa mendeteksi adanya keluaran pada data tersebut [2].

Pada laporan klasifikasi, terdapat model evaluasi yang terdiri dari empat metrik. Empat metrik tersebut berupa akurasi, presisi, nilai F1, dan *recall*. Akurasi adalah nilai ketepatan dalam pengujian data. Presisi adalah ukuran tingkat akurasi

algoritma model dalam mengidentifikasi contoh relevan dengan benar dalam jumlah total yang diimplementasikan secara positif. Nilai F1 adalah metrik akurasi pembelajaran mesin yang mengukur akurasi model. Sedangkan *recall* adalah metrik yang digunakan untuk mengukur efektivitas model. Keempat metrik ini digunakan untuk memperoleh nilai rata-rata ketepatan yang lebih akurat hingga 90-92% [28].

Hasil dari keempat metrik pada laporan klasifikasi dapat dijabarkan dengan menggunakan rumus Matematika. Rumus tersebut terdapat pada Formula 2.7 sampai Formula 2.10:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.8}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.9}$$

$$NilaiF1 = 2 \cdot [\frac{Precision \cdot Recall}{Precision + Recall}] \tag{2.10}$$

Berdasarkan Formula 2.7 [28], nilai akurasi akan lebih baik jika nilai positif asli dan nilai negatif asli lebih tinggi dibandingkan nilai positif palsu dan nilai negatif palsu. Pada Formula 2.8 [28], nilai presisi akan lebih baik jika nilai positif asli (TP) semakin tinggi dan nilai positif palsu (FP) semakin rendah. Pada Formula 2.9 [28], nilai *recall* akan lebih baik jika nilai positif asli semakin tinggi (TP) dan nilai negatif palsu semakin rendah (FN). Pada Formula 2.10 [28], nilai F1 akan lebih tinggi jika nilai presisi dan nilai *recall* mendekati 100%.

Nilai dari keempat metrik juga memiliki rata-rata makro dan rata-rata berbobot. Rata-rata makro merupakan rata-rata yang digunakan untuk mencari bobot setiap kelas tanpa memperhatikan seberapa besar atau kecilnya jumlah sampel dalam kelas tersebut. Rata-rata berbobot adalah perhitungan rata-rata nilai bobot pada sampel setiap kelas, kelas yang nilainya besar akan memengaruhi nilai rata-rata bobot tersebut [35]. Nilai rata-rata makro diperoleh dengan menggunakan Formula 2.11 dan Formula 2.12.

$$\frac{TP}{TP + 0.5 \cdot (FP + FN)} \tag{2.11}$$

$$\sum_{a}^{b} (PerclassF1score \cdot SupportPropotion) \tag{2.12}$$

Berdasarkan Formula 2.11 [35], nilai rata-rata makro akan mendekati 100% jika nilai positif asli semakin tinggi. Pada Formula 2.12 [35], nilai rata-rata makro diperoleh dengan melibatkan nilai F1 dan proporsi nilai dukungan. Pada konteks kali ini, batas nilai rata-rata makro dapat dihitung dengan menggunakan data array berapa pun dengan nilai a merupakan nilai array data terendah, dan nilai b merupakan nilai array data tertinggi [35].

2.7 Model h5

Penelitian yang berbasis pembelajaran mendalam, model perlu dibangun untuk melakukan klasifikasi emosi dan ekspresi gambar menggunakan CNN. Model tersebut memiliki banyak himpunan data dengan ukuran memori yang cukup besar. Kondisi ini akan membutuhkan waktu yang lama untuk menjalankan proses tersebut. Pada konteks kali ini, proses penyimpanan model sangat dibutuhkan. Model tersebut akan disimpan dengan format h5 [34].

Model h5 adalah semua *Hierarchical Data Format* (HDF) yang digunakan untuk menyimpan model. Model tersebut berupa hasil dari pengujian data, prediksi data, dan proses visualisasi gambar. Model ini akan disimpan secara otomatis pada program komputer setelah model tersebut dijalankan. Model ini sangat berguna pada penelitian yang berbasis pembelajaran mendalam, CNN, dan Python [34].

Model h5 yang disimpan berisi beragam jenis himpunan data. Himpunan data tersebut adalah himpunan data yang berada pada model pra-pelatihan. Model pra-pelatihan tersebut terdiri dari arsitektur CNN, optimalisasi Adam, jumlah iterasi, tingkat kerugian, dan tingkat akurasi data. Selain itu, hasil dari model juga akan diperoleh berdasarkan matriks *confusion* dan laporan klasifikasi [34]. Dengan demikian, proses menjalankan model pra-pelatihan tidak perlu dilakukan secara berulang.

Selain dari model h5, ada model yang bentuknya berupa gambar atau tulisan. Model yang bentuknya berupa gambar biasanya disimpan dengan format PNG atau JPG. Model dalam bentuk tulisan biasanya disimpan dengan format txt. Model dalam bentuk gambar seperti *confusion* matriks dan grafik lapisan pada CNN. Model dalam bentuk tulisan seperti akurasi data dan tingkat kerugian. Semua model yang tersimpan tersebut dimanfaatkan agar proses menjalankan model pelatihan dilakukan berkali-kali.