BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tujuan Pustaka

Tinjauan pustaka merupakan acuan utama dalam beberapa studi yang pernah dilakukan yang berkaitan dengan penelitian ini. Berikut ini merupakan beberapa penelitian yang digunakan berdasarkan metode MFCC dan algoritma klasifikasi yang digunakan sebagai acuan dalam penelitian ini.

Penelitian pertama yang dilakukan oleh (Tanuar *et al*., 2020). Dalam penelitian ini, bertujuan untuk melakukan identifikasi gender dalam Bahasa Indonesia menggunakan algoritma *supervised machine learning*. Data primer dikumpulkan dalam Bahasa Indonesia untuk mengidentifikasi gender dengan menggunakan MFCC sebagai algoritma ekstraksi fitur. Berbagai algoritma *machine learning* seperti *Artificial Neural Network*, SVM, dan *K-Nearest Neighbor* dibandingkan untuk mengevaluasi efektivitas mereka dalam identifikasi gender. Metode yang digunakan melibatkan pengumpulan data primer dari 50 wanita dan 52 pria yang merekam 8 kalimat perintah sederhana dalam Bahasa Indonesia. Total data yang digunakan adalah 2.735, dengan komposisi 1.296 data suara perempuan dan 1.493 data suara laki-laki. Hasil penelitian menunjukkan bahwa algoritma *Artificial Neural Network* (ANN) memberikan hasil yang lebih menjanjikan dibandingkan dengan algoritma lainnya seperti SVM dan *K-Nearest Neighbor*. ANN menunjukkan akurasi yang lebih tinggi dalam identifikasi gender suara dalam Bahasa Indonesia.

Penelitian kedua yang dilakukan oleh (Nashipudimath *et al*., 2021). Dalam penelitian ini, bertujuan untuk mengembangkan model pengenalan gender dan emosi melalui ekstraksi fitur suara. Model ini bertujuan untuk meningkatkan akurasi dalam mengidentifikasi gender dan emosi seseorang berdasarkan sinyal suara yang diterima. Selain itu, penelitian ini juga bertujuan untuk mengeksplorasi penggunaan teknik-teknik seperti *Voice Activity Detector, Mel-Frequency Cepstral Coefficient, Principal Component Analysis, dan Support Vector Machine classifier* dalam konteks pengenalan gender dan emosi melalui suara. Hasil penelitian menunjukkan bahwa pendekatan yang digunakan mampu menghasilkan hasil yang akurat. Dengan menggunakan teknik *Voice Activity Detector, Mel-Frequency Cepstral Coefficient, Principal Component Analysis,* dan *Support Vector Machine classifier*, penelitian ini berhasil dalam pengenalan gender dan emosi melalui analisis fitur suara.

Penelitian ketiga yang dilakukan oleh (Nasef, Sauber, *and*, Nabil 2021). Dalam penelitian ini, bertujuan untuk mengembangkan sistem pengenalan gender suara *end-to-end* yang efektif di lingkungan yang tidak terbatas. Metode yang digunakan melibatkan penggunaan *Mel-frequency cepstral coefficients* (MFCC) sebagai representasi data audio, dengan penerapan *self-attention* dalam dua model yang berbeda. Hasil penelitian menunjukkan bahwa kedua model yang diusulkan mampu mencapai tingkat akurasi yang tinggi, yaitu 95.11% dan 96.23%, serta menunjukkan kinerja unggul dalam kondisi lingkungan yang tidak terkontrol.

Penelitian keempat yang dilakukan oleh (Uddin *et al.*, 2020). Dalam penelitian ini, bertujuan untuk mengenali gender dari suara manusia menggunakan

MFCC yang efisien dalam mengekstrak fitur dari ucapan audio. Metode yang digunakan melibatkan *preprocessing* data untuk menghilangkan *noise* dan menggunakan algoritma KNN untuk klasifikasi gender. Hasil penelitian menunjukkan tingkat akurasi yang tinggi, seperti 96.8% untuk *dataset* TIMIT, 92.5% untuk *dataset* RAVDESS, dan 96.6% untuk *dataset* BGC menggunakan KNN. Metode yang diusulkan dalam penelitian ini berhasil menunjukkan peningkatan signifikan dalam akurasi pengenalan gender dari suara manusia.

Penelitian kelima yang dilakukan oleh (Ali *et al*., 2022). Dalam penelitian ini, bertujuan untuk membandingkan kinerja model berbasis LPC dan MFCC dalam sistem pengenalan gender berbasis ucapan. Metode penelitian melibatkan pengumpulan data suara dari 93 pembicara yang mengucapkan kata-kata tertentu dengan vokal yang berbeda, kemudian menganalisis data menggunakan teknik klasifikasi ANN. Hasilnya menunjukkan bahwa penggunaan MFCC-ANN memberikan akurasi tertinggi sebesar 97.07% dalam pengenalan gender, dengan kinerja hampir sama baiknya untuk kelas pria dan wanita.

Tabel 2.1 Tabel Tinjauan Pustaka

No	Nama Peneliti	Tujuan Penelitian	Metode Penelitian	Hasil Penelitian
1	Tanuar, E., Abdurachman, E., Gaol, F., Lukas,	Tujuan penelitian ini adalah untuk melakukan identifikasi gender dalam Bahasa Indonesia menggunakan algoritma *supervised machine learning*	MFCC, ANN, SVM, KNN.	Penelitian ini menunjukkan algoritma ANN memberikan hasil yang baik dalam identifikasi gender suara dibandingkan dengan algoritma SVM dan KNN

No	Nama Peneliti	Tujuan Penelitian	Metode Penelitian	Hasil Penelitian
2	Nashipudi math, M., Pillai, P., Subramani an, A., Nair, V., Khalife, S.	Tujuan penelitian ini adalah untuk mengidentifikasi gender dan emosi seseorang melalui ekstraksi fitur suara	*Voice Activity Detector,* MFCC, *Principal Component Analysis*, & SVM.	Penelitian ini menunjukkan akurasi sebesar 98.88% untuk pengenalan gender dan 72.02% untuk pengenalan emosi. Hasil ini dicapai melalui teknik
				pre-processing yang cermat menggunakan VAD, serta ekstraksi fitur yang lebih baik dengan MFCC dan PCA.
3	Nasef, M., Sauber, A., Nabil, M.	Tujuan penelitian ini adalah untuk mengembangkan sistem pengenalan gender suara yang efektif di lingkungan yang tidak terbatas menggunakan pendekatan *self-attention*.	MFCC*, Pure Self-Attention* dan *Convolution Self-Attention*	Penelitian ini menunjukkan bahwa kedua model yang diusulkan mencapai tingkat akurasi tinggi, yaitu 95.11% dan 96.23%. Model ini unggul dalam semua kriteria dan dianggap *state-of-the-art* untuk pengenalan gender suara di lingkungan yang tidak terbatas.
4	Uddin, M., Hossain, M., Pathan, R., Biswas, M.	Tujuan penelitian ini adalah untuk mengembangkan sistem pengenalan gender dari suara manusia menggunakan teknik *Machine Learning* dengan pendekatan MFCC untuk mengekstrak fitur-fitur yang representatif dari ucapan audio.	MFCC, LPC, KNN, GMM, RF, ANN, SVM	Penelitian menunjukkan bahwa pengenalan gender dari suara manusia dengan algoritma KNN memiliki tingkat akurasi tinggi, yaitu 96.8% pada *dataset* TIMIT, 92.5% pada *dataset* RAVDESS, dan 96.6% pada *dataset* BGC.

No	Nama Peneliti	Tujuan Penelitian	Metode Penelitian	Hasil Penelitian
5	Alnuaim, A., Zakariah, M., Shashidar, Cs., Hatamleh, W., Tarazi, H., Shukla, P., Ratna, R.	Tujuan dari penelitian ini adalah untuk membandingkan kinerja model berbasis *Linear Prediction Coefficients* (LPC) dan *Mel-Frequency Cepstral Coefficients* (MFCC) dalam pengenalan gender berbasis ucapan.	LPC & MFCC	Penelitian menunjukkan bahwa penggunaan MFCC-ANN mencapai akurasi tertinggi 97.07% dalam pengenalan gender berbasis ucapan, dengan kinerja yang hampir sama baiknya untuk pria dan wanita. MFCC secara konsisten lebih unggul dibandingkan fitur LPC dalam klasifikasi menggunakan DA dan ANN.

2.2 *Voice Recognition*

Menurut *Joseph P. Campbell* dalam jurnal yang ditulis (Mutohar, 2006), *voice recognition* adalah suatu proses untuk mengidentifikasikan seseorang dengan mengenali suara dari orang tersebut. *Voice Recognition* memiliki dua kategori yaitu *Text-Dependent* dimana sistem dilatih untuk mengenal frasa sandi suara (*voiceprint*) yang telah di tentukan, sedangkan *Text-Independent* sistem tidak memerlukan kata sandi suara (*voiceprint*) yang telah ditentukan sebelumnya. *Voice Recognition* atau pengenalan suara atau ucapan adalah suatu teknik yang memungkinkan sistem komputer untuk mengidentifikasi seseorang berdasarkan sandi suara, dengan memindai ucapan dan mencocokkan dengan sandi suara.

2.3 Sinyal Suara Manusia

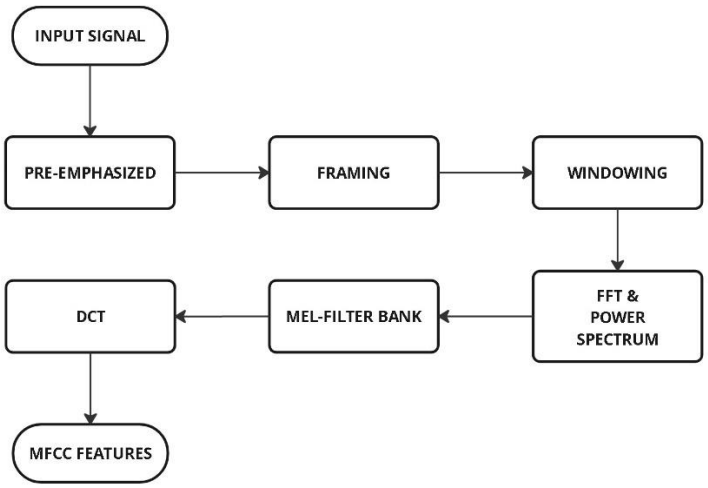
Suara manusia merupakan gabungan dari berbagai sinyal analog. Tetapi suara murni secara teoritis dapat dijelaskan dengan kecepatan osilasi atau frekuensi yang diukur dalam ukuran *Hertz* (Hz) dan amplitudo atau tingkat kejernihan bunyi dengan pengukuran dalam desibel. Suara ini mengalir dengan perantara udara. Gelombang suara di atas 20 kHz merupakan gelombang ultrasonik dan gelombang suara di bawah 20 Hz merupakan gelombang infrasonik.

Manusia menghasilkan suara melalui pita suara. Ketika manusia berbicara pita suara inilah yang bergetar sehingga dapat menghasilkan suatu bunyi atau suara. Suara digunakan oleh manusia untuk berkomunikasi dan berinteraksi dalam kehidupan sehari-hari (Anggraini *and* Fadillah, 2019).

2.4 *Mel Frequency Cepstral Coefficient* (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) adalah representasi fitur yang umum digunakan dalam pengolahan sinyal suara, terutama dalam pengenalan suara dan analisis audio. MFCC dirancang untuk meniru cara manusia mendengar suara; dalam jurnal yang ditulis (Prabakaran *and* Sriuppili, 2021) *Mel-Frequency Cepstrum Coefficient* (MFCC) adalah koefisien yang bersama-sama membentuk MFCC, koefisien ini berasal dari jenis representasi *cepstral* dari klip audio. Didalam MFCC, pita frekuensi berjarak sama pada *Mel-scale*, yang mendekati respons sistem pendengaran manusia. Frekuensi ini dapat menghasilkan representasi suara yang lebih baik, misalnya, dalam kompresi audio.

Teknik ini digunakan untuk melakukan *feature extraction* dari audio, sebuah proses yang mengonversikan sinyal suara menjadi beberapa parameter. Seperti pada Gambar 2.1 tahapan untuk melakukan *feature extraction* suara dimulai dari *input signal*, selanjutnya dilakukan proses *Pre-emphasize, Framing, Windowing, Fast Fourier Transform, Mel Filter Bank,* dan *Discrete Cosine Transform* lalu menghasilkan MFCC *features*



Gambar 2.1 Tahapan MFCC

2.4.1 *Pre-Emphasized*

Pre-emphasis adalah langkah pertama dalam ekstraksi fitur suara atau sinyal. Langkah ini bertujuan untuk memperkuat komponen frekuensi tinggi dari sinyal input, ini dilakukan karena sinyal suara alami biasanya memiliki lebih banyak energi di frekuensi rendah, sehingga sinyal frekuensi tinggi mungkin kurang terepresentasi (Wang, Fu, *and* Abza, 2024). Seperti pada Gambar 2.2. Pre-emphasis dapat dihitung dengan persamaan (2.1).

$$y[n] = x[n] - \alpha . x\,[n-1] \tag{2.1}$$

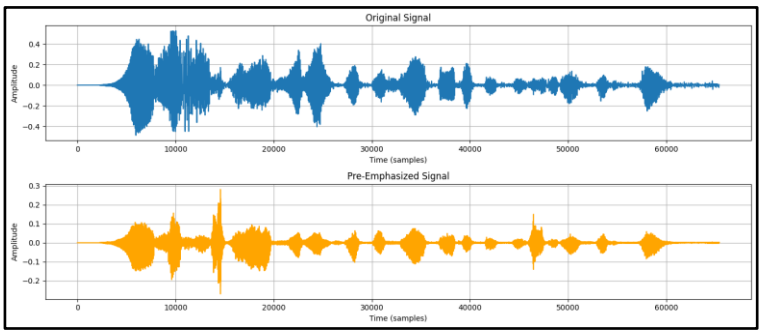
Keterangan :

$y[n]$: Sinyal hasil *pre-emphasize filter*,

$x[n]$: Sinyal sebelum *pre-emphasize filter*,

α : Nilai alfa.

Pada umumnya nilai α (*alfa*) yang sering digunakan adalah antara 0,9 sampai 1.0, sedangkan *default* nilai α sendiri adalah 0.97 (Indrawaty, Dewi, *and* Lukman, 2019).



Gambar 2.2 *Signal Pre-Emphasized*

2.4.2 *Framing*

Framing adalah proses membagi sinyal suara menjadi beberapa frame untuk menangkap informasi spektral lokal. Sinyal suara dibagi menjadi frame-frame dengan durasi antara 20 hingga 30 ms. Setiap frame terdiri dari N sampel, dan frame-frame ini saling tumpang tindih dengan jarak antar frame sebesar M sampel, di mana M umumnya sebesar 50% dari panjang frame. Ukuran frame biasanya dipilih sekitar 20–30 ms, dengan tumpang tindih (*overlap*) 50% (Wang, Fu, *and* Abza, 2024). Tumpang tindih ini dilakukan untuk memastikan kontinuitas informasi antara frame yang berdekatan dan menghindari hilangnya informasi penting pada batas-batas frame, seperti yang terlihat pada Gambar 2.3. Penghitungan *framing* dapat menggunakan persamaan (2.2).

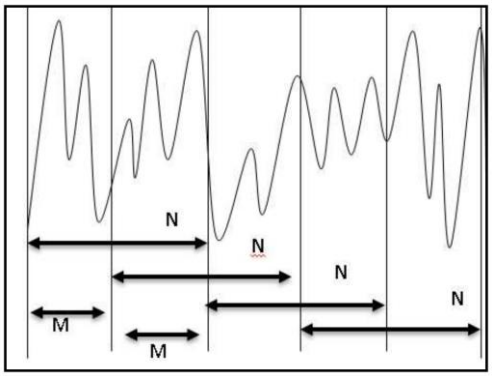
$$x_f[n] = x[n] + f \cdot (N - M) \quad (2.2)$$

Keterangan :

N : panjang frame dalam sampel.

M : jumlah sampel yang dipindahkan antara frame berurutan.

f : indeks frame.

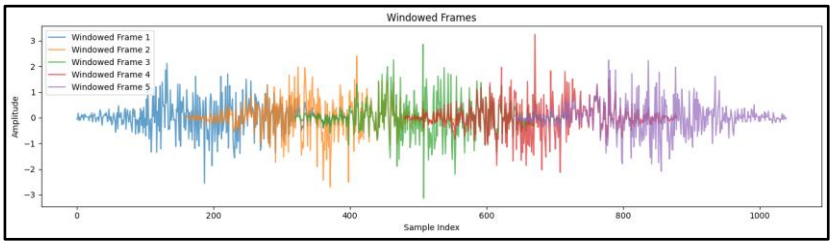


Gambar 2.3 *Framing Signal Process*

(Sumber: (Heriyanto, Hartati, and Eko, 2018))

2.4.3 *Windowing*

Hasil dari proses *Framing* menghasilkan efek sinyal diskontinuitas, agar tidak terjadi kesalahan data pada proses *fourier transform* maka sampel suara yang telah dibagi menjadi beberapa *frame* perlu dijadikan sinyal *continue* dengan menggunakan proses *windowing*. Seperti pada Gambar 2.4 Tahap windowing mengalikan setiap *sample* di dalam *frame* dengan sebuah fungsi *window* untuk mengurangi efek diskontinuitas atau kesenjangan pada bagian awal dan akhir setiap *frame* (Permana, Fiolana, *and* Diah, 2022).



Gambar 2.4 *Signal Windowing*

Ada banyak fungsi *windowing* namun fungsi yang sering digunakan dari *windowing* adalah *hamming window* seperti terlihat pada Gambar 2.5 *Hamming Window* diperlukan untuk mengurangi efek diskontinuitas dari proses *frame blocking* terutama pada ujung awal dan ujung akhir setiap frame (Nursholihatun, Sasongko, *and* Zainuddin, 2020). Penghitungan *hamming window* dapat menggunakan Persamaan (2.3).

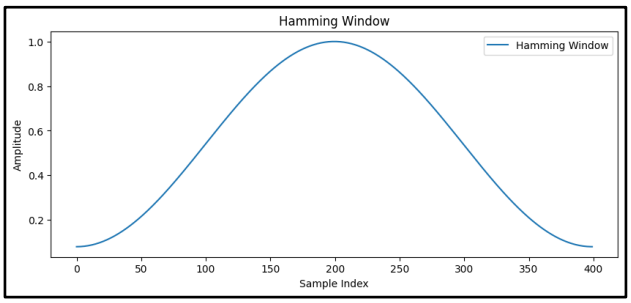
$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \tag{2.3}$$

Keterangan :

$w(n)$: *Hamming window,*

n : 0,1,2, …, M-1,

N : Panjang *frame.*



Gambar 2.5 *Signal Hamming Window*

2.4.4 *Fast Fourier Transform* (FFT)

Fast Fourier Transform (FFT) adalah teknik perhitungan operasi matematika yang digunakan untuk mentrasformasi sinyal analog berbasis waktu menjadi sinyal digital berbasis frekuensi. *Fast Fourier Transform* (FFT) membagi sebuah sinyal menjadi frekuensi yang berbeda-beda dalam fungsi eksponensial yang kompleks (Kusuma, 2020). Penghitungan FFT dapat menggunakan Persamaan (2.4).

$$X(k) = \sum_{n=0}^{N-1} x_w\,[n] \cdot e^{-j2\pi kn/N} \tag{2.4}$$

Keterangan :

$X(k)$: *Fourier Form Transform*

$x_w[n]$: Nilai sinyal input setelah di-window

N : Panjang atau jumlah total sampel dalam satu frame atau segmen sinyal

n : Indeks waktu dalam domain waktu

k : Indeks frekuensi dalam domain frekuensi

j : Bilangan imajiner

Setelah diubah kedalam frekuensi, output yang didapatkan nantinya adalah spektrum, untuk memahami karakeristik dari frekuensi sinyal maka dilakukan proses *power spectrum* dengan Persamaan (2.5).

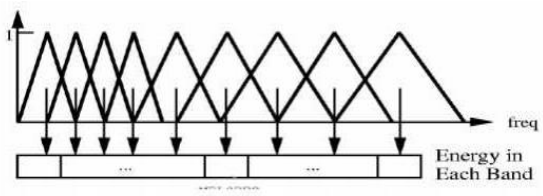
$$P[k] = \frac{1}{N}|X[k]|^2 \tag{2.5}$$

$P[k]$: Daya sinyal pada *bin* frekuensi ke-*k.*

$X[k]$: Nilai kompleks dari *Fast Fourier Transform* (FFT)

2.4.5 *Mel Filter Bank*

Filter bank adalah salah satu bentuk dari filter yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari frekuensi *band* tertentu dalam sinyal suara. *Filter bank* menggunakan representasi konvolusi dalam melakukan penyaringan terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koefisien *filter bank* (Candra, 2021).



Gambar 2.6 *Mel Scale Filter Bank*

(Sumber: Sable *et al*., 2013)

Proses *Filter bank* pada Gambar 2.6 menunjukan filter segitiga yang digunakan untuk menghitung jumlah filter komponen spektral sehingga *output* dari proses mendekati *Mel-scale. Mel-scale* bertujuan untuk meniru persepsi suara telinga manusia non-linear, dengan menjadi lebih diskriminatif pada frekuensi yang lebih rendah dan kurang deskriminatif pada frekuensi yang lebih tinggi. Pada *filter bank* dapat mengkonversi antara *Hertz* ke *Mel* dengan Persamaan (2.6) maupun sebaliknya dengan Persamaan (2.7) (Ridho, 2019).

Hertz ke *Mel* :

$$F_{mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.6}$$

Keterangan :

F_{mel} : Koefisien *filter bank,*

f : Frekuensi

Mel ke *Hertz* :

$$F_{hz} = 700\left(10^{\frac{F_{mel}}{2595}} - 1\right) \tag{2.7}$$

F_{hz} : Frekuensi dalam *Hertz,*

F_{mel} : Frekuensi dalam *Mel*

Selanjutnya *filter bank* dimodelkan sesuai persepsi telinga manusia terhadap suara dengan filter segitiga. Filter ini saling tumpang tindih, memastikan bahwa seluruh rentang frekuensi tercakup dan bahwa setiap komponen frekuensi dihaluskan pada rentang frekuensi tertentu. Perhitungan pemodelan filter segitiga menggunakan Persamaan (2.10).

$$H_m(k) = \begin{cases} 0 & \text{if} k < f_{m-1} \\ \dfrac{k - f_m - 1}{f_m - f_m - 1} & \text{if} f_{m-1} \le k < f_m \\ \dfrac{f_m + 1 - k}{f_m + 1 - f_m} & \text{if} f_m \le k < f_m + 1 \\ 0 & \text{if} k > f_{m+1} \end{cases} \tag{2.10}$$

Keterangan :

$H_m(k)$: Nilai filter segitiga ke-*m* pada *bin* frekuensi ke-*k*

k : Indeks bin frekuensi

f_{m-1} : Batas frekuensi bawah dari filter segitiga ke-*m*

f_m : Frekuensi pusat dari filter segitiga ke-*m*

2.4.6 *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform (DCT) adalah langkah terakhir ekstraksi fitur dengan MFCC. DCT juga mengorelasikan *mel spectrum* sehingga menghasilkan representasi yang baik dari *property* spektral lokal. Pada proses ini nilai *mel*

spectrum dalam *frequency domain* akan di konversikan kedalam *time domain* (Candra, 2021). Penghitungan DCT dapat menggunakan Persamaan (2.11).

$$C_k = \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{\pi}{N} \cdot \left(n + \frac{1}{2}\right) \cdot k\right) \tag{2.11}$$

Keterangan :

C_k : koefisien MFCC ke-n

$x[n]$: energi *log* dari *Mel Filter Bank* ke-m.

N : jumlah total *Mel Filter Bank*.

k : indeks koefisien MFCC yang dihitung, berkisar dari 0 hingga N−1.

2.5 *Normalization*

Normalisasi dalam konteks MFCC, khususnya *Cepstral Mean Normalization* (CMN), adalah proses yang dirancang untuk mengurangi variabilitas dalam sinyal suara yang disebabkan oleh kondisi perekaman dan saluran yang berbeda. CMN mencapai ini dengan memusatkan koefisien MFCC di sekitar nol, sehingga mengurangi dampak distorsi saluran *linier* dan *noise* statis. Ini dicapai dengan menghitung rata-rata koefisien MFCC selama segmen bingkai audio. Manfaat utama dari CMN adalah kemampuannya untuk meningkatkan ketahanan dan keandalan sistem pengenalan suara dan pembicara, yang mengarah pada peningkatan akurasi (Kalinli, Bhattacharya, *and* Weng, 2019). Sebelum melakukan perhitungan normalisasi menggunakan Persamaan (2.13) , perhitungan rata-rata MFCC harus dilakukan terlebih dahulu dengan Persamaan (2.12).

Menghitung rata-rata :

$$\mu = \frac{1}{N} \sum_{i=1}^{N} C_i \tag{2.12}$$

μ : rata-rata dari koefisien MFCC

C_i : mewakili koefisien MFCC ke-i

N : jumlah total koefisien MFCC

Normaslisasi rata-rata :

$$C_i' = C_i - \mu \tag{2.13}$$

C_i' : koefisien MFCC yang telah di normalisasi

C_i : koefisien MFCC yang asli

μ : rata-rata dari koefisien MFCC

2.6 *Padding*

Padding dalam proses MFCC adalah teknik yang digunakan untuk menstandarkan panjang vektor fitur yang diekstraksi dari sinyal audio, yang sangat berguna saat menangani panjang *input* audio yang bervariasi. Proses ini memastikan bahwa semua vektor fitur MFCC memiliki panjang yang sama, sehingga cocok untuk pemrosesan lebih lanjut dan sebagai *input* ke model pembelajaran mesin (Singh *et al.*, 2021).

Tujuan penerapan *padding* pada MFCC adalah untuk memotong urutan MFCC yang lebih panjang dari panjang yang ditentukan atau memperpanjang urutan yang lebih pendek agar sesuai dengan panjang yang diinginkan. Ini sering dilakukan dengan menambahkan nol di akhir vektor MFCC, metode yang dikenal sebagai *zero-padding*. *Zero-padding* hanya memperpanjang vektor ke panjang yang seragam (Singh *et al.,* 2021). Berikut poin penting dari proses *padding*:

1. *Consistency* : *Padding* memastikan bahwa semua vektor *input* ke model pembelajaran mesin memiliki panjang yang sama, yang merupakan persyaratan untuk banyak model.
2. *Zero padding* : Ini melibatkan penambahan nol di akhir vektor MFCC yang lebih pendek. Ini adalah metode sederhana dan efektif untuk menangani *input* dengan panjang yang bervariasi.
3. *Truncation* : Untuk vektor MFCC yang lebih panjang dari panjang yang diinginkan, trunkasi menghapus elemen-elemen tambahan, memastikan semua vektor memiliki panjang yang sama.

2.7 *Convolutional Neural Network* (CNN)

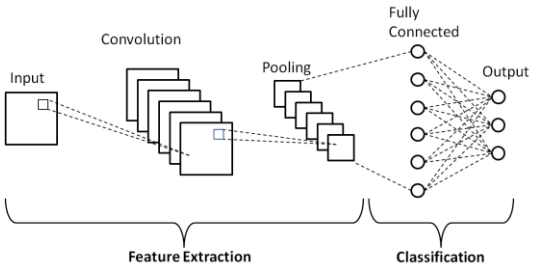
Model *deep learning* dapat dipandang sebagai sebuah kelas model *neural network* dengan arsitektur yang lebih kompleks dan jumlah layer yang lebih banyak dari model *neural network* biasa, model *deep learning* memiliki fungsi khusus, salah satu yang merupakan algorima dari *deep learning* yaitu *convolutional neural network* (CNN).

CNN sangat berbeda dari algoritma pengenalan pola lainnya karena CNN menggabungkan ekstraksi fitur dan klasifikasi. Seperti pada Gambar 2.5.1 *Process CNN* menunjukkan contoh representasi skematik sederhana dari CNN dasar. Jaringan sederhana ini terdiri dari lima lapisan berbeda: lapisan *input*, lapisan *convolution*, lapisan *pooling*, lapisan *fully-connected*, dan lapisan *output*.

Lapisan-lapisan ini dibagi menjadi dua bagian: ekstraksi fitur dan klasifikasi. Ekstraksi fitur terdiri dari lapisan *input*, lapisan *convolution*, dan lapisan *pooling*,

sedangkan klasifikasi terdiri dari lapisan *fully-connected* dan lapisan output. Lapisan *input* menentukan ukuran tetap untuk *input* data, Kemudian, *input* data dikonvolusi dengan beberapa kernel yang dipelajari menggunakan bobot bersama oleh lapisan *convolution*. Selanjutnya, lapisan *pooling* mengurangi ukuran *input* data sambil mencoba mempertahankan informasi yang terkandung di dalamnya.

Output dari ekstraksi fitur dikenal sebagai peta fitur. Klasifikasi menggabungkan fitur yang diekstraksi di lapisan *fully-connected*. Terdapat satu *neuron output* untuk setiap kategori objek di lapisan *output*. *Output* dari bagian klasifikasi adalah hasil klasifikasi (Phung *and* Rhee, 2019).



Gambar 2. 7 *Process* CNN

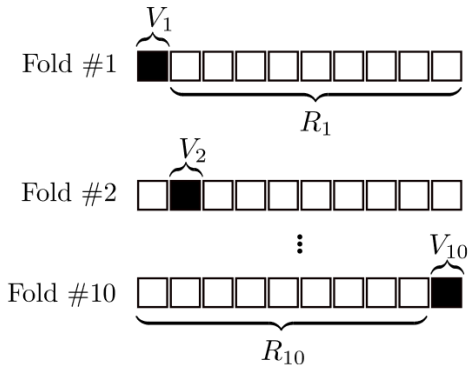
(Sumber : Phung *and* Rhee 2019)

2.8 *K-fold Cross Validation*

K-fold cross-validation adalah cara yang efektif untuk menggabungkan pemilihan fitur untuk melatih model prediksi yang optimal. Dalam pendekatan *K-fold cross-validation*, data dibagi menjadi *k* lipatan luar. Kemudian, lipatan dalam dibuat di setiap set pelatihan luar untuk memilih fitur, menyetel parameter, dan melatih model (Nugroho *and* Amrullah, 2023).

Dalam *k-fold cross-validation*, set pelatihan dibagi menjadi *k subset* yang tidak tumpang tindih dengan ukuran yang hampir sama. *"Fold"* mengacu pada

jumlah *subset* ini. Pembagian dilakukan dengan pengambilan sampel acak dari set pelatihan tanpa penggantian. Model dilatih pada *k−1 subset* dan diuji pada *subset* yang tersisa, disebut set validasi. Proses ini diulang sehingga setiap *subset* menjadi set validasi sekali. Rata-rata kinerja dari *k* validasi adalah kinerja *cross-validated*. Gambar 2.8 mengilustrasikan 10-*fold cross-validation*, di mana pada lipatan pertama, *subset* pertama adalah set validasi *V1* dan sisanya adalah set pelatihan *R1*. Pada lipatan kedua, *subset* kedua menjadi set validasi, dan seterusnya. (Berrar, 2024).



Gambar 2. 8 *K-fold cross validation*

(Sumber :Berrar, 2024)

2.9 *Python*

Python adalah bahasa pemrograman tingkat tinggi yang dikenal karena sintaksnya yang sederhana dan mudah dibaca, yang membuatnya populer di kalangan pemula maupun ahli. *Python* digunakan secara luas dalam berbagai bidang seperti pengembangan web, analisis data, kecerdasan buatan, dan komputasi ilmiah. Salah satu kekuatan utama *Python* adalah ekosistemnya yang kaya dengan berbagai pustaka dan kerangka kerja seperti *NumPy*, *Pandas*,

TensorFlow, dan *Django*, yang memudahkan pengembang untuk membangun aplikasi yang kompleks dengan lebih efisien. Selain itu, komunitas *Python* yang besar dan aktif terus berkontribusi pada pengembangan bahasa ini dengan berbagai sumber daya dan dokumentasi yang tersedia secara *online* (Rossum Guido van *and the Python development team*, 2017).

2.10 *VoxCeleb*

VoxCeleb adalah *dataset* yang kaya dan luas digunakan dalam penelitian pengenalan suara. *Dataset* ini terdiri dari rekaman audio wawancara selebriti yang dikumpulkan dari *platform* seperti *youtube*. *VoxCeleb* dibuat untuk mendukung penelitian dibidang suara (*speaker recognition*), verifikasi suara (*speaker verification*) dan tugas-tugas terkait lainnya. *Dataset* ini sering digunakan dalam pengembangan dan evaluasi sistem identifikasi dan verifikasi *speaker* otomatis.

VoxCeleb berisi rekaman suara dari wawancara selebriti yang dikumpulkan secara otomatis dari *youtube* menggunakan metode *crawling*. Vidio-vidio tersebut kemudian diolah untuk mengekstak audio, yang kemudian diiris menjadi segmen-segmen suara individu.

2.11 Flask

Flask merupakan web *framework* yang ditulis dalam bahasa pemograman *python*, *Flask* merupakan jenis *microframework* yang tidak memerlukan *library* tertentu dalam penggunaanya. *Flask* dapat menggunakan ektensi untuk menambahkan fitur atau komponen yang sudah disediakan oleh pihak ketiga dan tidak dipasang secara standar pada *flask* (Novindri *and* Ocsa, 2022).