SPEECH TO TEXT BAHASA SASAK MENGGUNAKAN EXTRAKSI FITUR *MEL-FREQUENCY CEPSTRAL COEFFICIENTS DAN* KLASIFIKASI *CONVOLUTIONAL NEURAL NETWORKS*

(*Speech to Text Sasak Language Using Mel-Frequency Cepstral Coefficients Feature Extraction and Convolutional Neural Networks Classification***)**

Belmiro Razak Setiawan, Arik Aranta*, Budi Irmawati
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia
Email: razak.haha@gmail.com, [arikaranta, budi-i] @unram.ac.id

Abstract

Artificial intelligence technology allows digital signals to be processed by computers. Currently speech to text is available only in Indonesian and English versions. Speech to text is a system that performs commands from human voice input os si then translated into words. The development of speech to text in regional languages is needed because it can be a bridge between culture and technological progress. From 5 research literature, it was found that the mel-frequency cepstral coefficients (MFCC) and convolutional neural networks (CNN) methods are a combination of the commonly used voice signal analysis methods and get accuracy between 70.00% to 99.00%. This study uses the CNN and MFCC methods in the speech to text field to recognize the Sasak language and convert it into text. The result of this research is a real time conversion system from voice to text in Sasak language. The analysis carried out includes determining the best amount of training data, testing the training data on the number of votes based on accuracy, the sensitivity of the algorithm to words that have similar prefixes using the MFCC method as feature extraction and CNN as a classifier for the voice dataset. This study aims to obtain the accuracy of the dataset used and the sensitivity of the algorithm to sentences that have similarities. In this study got 2 results. The first result is the result of training with the accuracy of CNN training is 90% and os sis 0.5%. The second result is the result of an experiment using 3 voice samples for each word on the dateset with 43 correct words, 6 correct words 2, 1 correct word 1 and none of the words incorrect. So it has a percentage of 86% all correct, 12% correct 2, and 2% correct 1, and 0% all wrong.

Keywords: *Voice, Sasak language, Speech to text, MFCC, CNN,*

**Penulis Korespondensi*

1. PENDAHULUAN

Perkembangan sistem *speech* pada masa ini mulai berkembang dengan pesat. Perkembangan tersebut terlihat dari banyaknya teknologi yang menggunakan sistem *speech* baik dalam bentuk perangkat keras maupun perangkat lunak dan ditambah lagi, banyak peneliti yang membuat jurnal mengenai sistem *speech*. Di Indonesia pun sama, sistem *speech* banyak digunakan sebagai perangkat lunak di handphone dan penelitian terhadap sistem *speech* mulai banyak yang mengembangkannya[1]. Bahasa pada sistem *speech* pun beragam dan salah satunya ada bahasa indonesia yang dikembangkan oleh google, dan kini banyak penelitian di indonesia yang mulai mengembangkan sistem *speech* menggunakan bahasa daerah.

Salah satu contoh aplikasi yang ada pada sistem *speech* adalah *speech to text*. *Speech to text* merupakan sistem yang mengerjakan perintah dari input suara manusia dan kemudian diterjemahkan menjadi teks yang sesuai dengan apa yang diucapkan[1]. *Speech to text* kini sudah banyak digunakan di kehidupan sehari-hari seperti google translate dan subtitle. Manfaat pada *speech to text* adalah untuk membuat sistem cerdas yang dapat mengenali suara manusia dan merubahnya menjadi teks agar manusia dapat mengetahui apa yang dikatakan orang tersebut.

Bahasa Sasak merupakan salah satu dari tiga bahasa daerah yang berasal dari Nusa Tenggara Barat yang paling banyak digunakan[2]. Bahasa Sasak digunakan oleh orang dengan etnis sasak yang kebanyakan berada di pulau Lombok. Bahasa sasak memiliki tiga tingkatan bahasa yaitu bahasa sasak halus, sedang dan biasa[3]. Dalam mendukung pelestarian bahasa

daerah, penggunaan bahasa daerah khususnya sasak harus dilakukan, salah satu upaya yang dapat dilakukan adalah dengan menggunakan pendekatan pengaplikasian teknologi dan kecerdasan buatan.

Alasan kenapa *speech to text* dibutuhkan karena *speech to text* dapat mempermudah manusia untuk mengetahui apa yang dikatakan oleh pembicara dengan cara membaca apa yang dikatakan pembicara. Tak hanya itu, *Speech to text* juga dapat digunakan untuk merubah suara menjadi kata atau kalimat dan hasil keluaran tersebut dibaca oleh komputer dan diubah menjadi suara lagi atau sinyal. Sehingga *speech to text* dapat digunakan dalam berbagai bidang seperti subtitle, *smart home* dan lain lain. Dengan penelitian ini dapat membantu pengembangan teknologi pada daerah terpencil atau orang-orang yang tidak bisa bahasa lain selain sasak. Sehingga daerah atau orang tersebut dapat terbantu dalam menggunakan perangkat *speech to text* dan dapat juga mengembangkan dan memperkenalkan budaya sasak dengan *speech to text*.

Dari 5 penelitian yang ditemukan terdapat akurasi yang bervariasi. Penelitian tersebut ada yang menggunakan CNN dan MFCC pada klasifikasi genre musik dengan akurasi optimal 99.00%[4], klasifikasi COVID-19 berdasarkan suara batuk dengan akurasi 70.58%[5], klasifikasi emosi dengan akurasi kelas 75.10% dan akurasi keseluruhan 76.10%[6], deteksi *asphyxia* berdasarkan suara tangisan dengan akurasi training 94.30% dan akurasi test 92.80%[7], deteksi emosi jangka panjang dengan akurasi 70.00%[8], dan penelitian *speech to text* terhadap kosa kata bahasa arab menggunakan metode MFCC dan HMM dengan akurasi 83.1% pada data sampling di frekuensi 8000z, 82.3% pada data sampling di frekuensi 22050Hz dan 82.2% pada data sampling di frekuensi 44100Hz[9].

Berangkat dari penelitian tersebut maka penelitian ini menggunakan metode CNN dan MFCC pada proses konfersi *speech to text* untuk mengenali bahasa sasak secara *real time* kedua metode ini dipilih dikarenakan Metode MFCC memiliki sifat menghasilkan fitur menggunakan mell yang dirasa dapat melakukan pengenalan pada setiap perbedaan Mel pada setiap kelas data, sedangkan CNN dipilih dikarenakan CNN bekerja baik pada data fitur suara yang cenderung memerlukan komputasi tingkat tinggi, sifatnya yang bekerja baik pada data dalam skala besar membuat metode CNN dipilih dalam perancangan penelitian ini. Hasil penelitian ini adalah akan dilakukan proses perancangan dan analisis penggunaan metode MFCC dan CNN untuk melakukan konfersi dari suara bahasa sasak menjadi teks meliputi penentuan jumlah data training terbaik, perbandingan data training dan data testing, kepekaan algoritma terhadap kata yang memiliki awalan yang serupa menggunakan metode MFCC sebagai fitur ekstraksi dan CNN sebagai klasifikasi terhadap dataset suara. Alasan menggunakan MFCC adalah karena tiap suara memiliki aplitudo dan durasi waktu yang dimana tiap suara memiliki aplitudo yang berbeda-beda. Sehingga MFCC merupakan metode yang tepat dalam mendeteksi perbedaan suara. proses yang dilakukan adalah dengan cara mengumpulkan suara yang memiliki kata yang sama kemudian dilabelkan. Dari pelabelan tersebut dapat dimaksimalkan menggunakan CNN. Pada CNN, hasil MFCC dilatih menggunakan CNN agar dapat mengetahui suara tersebut termasuk ke dalam label yang sesuai.

2. TINJAUAN PUSTAKA

Terdapat banyak penelitian yang terkait dengan *speech recognition* dan metode MFCC dan CNN. Dari penelitian tersebut dapat dijadikan acuan pada penelitian ini. Dari referensi yang didapatkan memiliki hasil akurasi yang besar.

Penelitian pertama berjudul "*MFCC-Based Feature Extraction Model for Long Time Period Emotion Speech Using CNN*"[8]. Penelitian tersebut menggunakan lima emosi dasar yaitu suka, sedih, senang, marah dan jijik sebagai penentunya. *Dataset* yang digunakan menggunakan *dataset* dari RAVDESS dan SAVEE berbasis FFT dengan total 400 hingga 1000 rekaman. *Dataset* tersebut diekstrak menjadi MFCC kemudian spektogram dari kata-kata ucapan jangka panjang diterapkan menjadi untuk mendapatkan hasil pembelajaran CNN. Akurasi yang didapatkan pada penelitian tersebut sebesar 70%.

Penelitian kedua berjudul "*Detection of asphyxia in infants using deep learning convolutional neural network (CNN) trained on Mel frequency cepstrum coefficient (MFCC) features extracted from cry sounds*"[7]. Penelitian tersebut menggunakan 284 sinyal *asphyxia* dan 316 sinyal tangisan biasa. Penelitian tersebut menghasilkan akurasi 94.29% akurasi pelatihan dan 92.78% pada akurasi uji coba. Dari penelitian tersebut MFCC yang dihasilkan dari tangisan bayi dapat digunakan pada CNN.

Penelitian ketiga berjudul "*Deep learning based emotion recognition system using speech features*

and transcriptions"[6]. Pada penelitian ini menggunakan MFCC untuk membantu mempertahankan karakteristik tingkat rendah yang berhubungan dengan emosi dalam ucapan dan menggunakan teks untuk menangkap makna semantik sehingga keduanya dapat membantu mendeteksi berbagai aspek dalam mendeteksi emosi. Dari kombinasi MFCC dan teks tersebut diuji coba pada beberapa *Deep Neural Network* (DNN). Dari semua metode tersebut, penelitian ini menggunakan menggunakan data IEMOCAP dan dapat membuktikan gabungan MFCC dan CNN memiliki keakuratan kelas 75.10% dan akurasi keseluruhan 76.10%.

Penelitian keempat berjudul "*Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks*"[5]. Pada penelitian ini mengklasifikasikan COVID-19 dengan menggunakan dataset batuk terbuka. Terdapat 2 label yang dilabelkan secara manual yaitu kelas COVID-19 dan non-COVID-19. *Audio* tersebut diubah menjadi MFCC dan dimasukan ke CNN. Akurasi yang didapatkan adalah 70.58% dengan sensitivitas 81% dan lebih baik dari pendekatan berbasis spektogram.

Penelitian kelima berjudul "Klasifikasi Genre Musik Menggunakan Metode *Deep Learning Convolutional Neural Network* dan *Mel-Spektrogram*"[4]. Pada penelitian ini menggunakan 10 jenis genre musik. Metode yang digunakan adalah MFCC dan kemudian MFCC tersebut diklasifikasi dengan CNN yang akan dibedakan activation funtionnya yaitu ReLU dan ELU. Dari hasil uji yang didapatkan mendapatkan akurasi paling tinggi yaitu 99%.

Penelitian keenam berjudul "Metode Mel Frequency Cepstral Coeffisients (MFCC) Pada klasifikasi Hidden Markov Model (HMM) Untuk Kata Arabic pada Penutur Indonesia"[9]. Pada penelitian ini menggunakan speech to text terhadap penuturan kosa kata bahasa arab berdasarkan dialek yang berbeda berdasarkan kecocokan suara dengan teks bahasa arab yang terdapat pada Al-Qur'an. menggunakan 3000 contoh suara yang dimana 150 contoh suara ini terdiri dari 5 orang indonesia asli yang tidak fasih bahasa arab dan 5 orang indonesia asli yang fasih berbahasa arab. Setiap penutur mengucapkan 15 kata dalam bahasa arab dan mengatakannya sebanyak 20 kali. Metode yang digunakan untuk ekstraksi fitur suara adalah MFCC dan metode untuk klasifikasi menggunakan HMM. Hasil uji yang didapatkan adalah akurasi 83.1% pada data sampling di frekuensi 8000z, 82.3% pada data sampling di frekuensi 22050Hz dan 82.2% pada data sampling di frekuensi 44100Hz.

Dari penelitian yang dijelaskan sebelumnya menjelaskan bahwa kombinasi MFCC dan CNN merupakan kombinasi metode yang umum digunakan dari metode yang lainnya. Dengan begitu, kombinasi MFCC dan CNN memungkinkan untuk dicoba pada penelitian *speech recognition* dengan bahasa sasak. Pada penelitian ini digunakan untuk mengenali kata atau kombinasi kata yang diucapkan yang diucapkan oleh user yang terdiri dari 50 kata.

2.1. Suara.

Suara merupakan gelombang longitudinal yang merambat melalui medium tertentu yang terjadi karena adanya getaran sehingga tercipta sebuah sistem suara yang pada akhirnya akan diterima oleh telinga[10]. Suara digunakan sebagai alat komunikasi yang digunakan pada hewan dan manusia. Manusia menggunakan suara sebagai alat komunikasi dengan cara mengirimkan pesan melalui suara dan akan diterima oleh telinga pendengar.

Pada umumnya, suara dapat digitalisasi menjadi gelombang analog. Frekuensi suara merupakan jumlah gelombang suara yang diterima oleh telinga pada setiap detiknya dengan satuan Herz (Hz)[10]. Frekuensi suara yang dapat diterima oleh manusia berkisar 16-20.000 Hz dan suara yang dihasilkan manusia berkisar 250-3000 Hz. Benda yang menghasilkan suara dengan frekuensi tertentu akan menjadi ciri khas dari benda tersebut.

2.2. *Speech to text Conversion*.

Speech to text conversion adalah metode yang digunakan untuk memecahkan kode suara manusia agar dapat dibaca oleh komputer dan dijadikan teks[9]. *speech to text conversion* memiliki langkah-langkah untuk dapat dibaca oleh komputer yaitu :.

2.2.1. *Pre-emphasis*

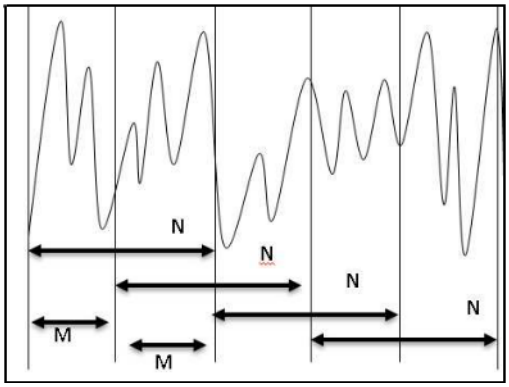
Pre-emphasis merupakan pemrosesan filter signal sederhana yang digunakan untuk mengurangi kebisingan (*noise*) untuk memperbaiki *Signal to Noise Rasio* (RNS)[9]. *Pre-emphasis* bertujuan untuk menjaga kualitas *level base band* pada frekuensi tinggi agar tetap memiliki kualitas sinyal yang baik. *Pre-emphasis* memiliki persamaan yang dapat dilihat pada Persamaan (1) yaitu[10]:

$$y(n) = s(n) - \alpha s(n-1) \quad (1)$$

Dimana y(n) merupakan Sinyal setelah *pre-emphasis*, s(n) merupakan Sinyal sebelum *pre-emphasis*, n merupakan nomor urut sinyal, α merupakan konstanta *filter pre-emphasis* di antara 0,9 sampai 1,0 (0,9 ≤ α ≤ 1,0) dan s merupakan Sinyal.

2.2.2. *Frame Blocking*

Frame blocking merupakan proses analisis sinyal ucapan dalam bentuk *frame*[11]. Setiap frame diwakili oleh fitur vektor tunggal digambarkan dalam spektrum rata-rata untuk interval waktu dalam *frame* diambil antara 20-40 milidetik [11]. Cara melakukan *frame blocking* adalah dengan pemotongan sinyal dan dibagi menjadi beberapa frame dengan masing-masing *frame* memuat N sampel sinyal dan *frame* yang berdekatan dipisah sejauh M sampel sehingga N = 2M dan M<N[11]. *Frame* diambil sepanjang mungkin untuk mendapatkan resolusi yang baik. Sedangkan waktu diambil sependek mungkin untuk mendapatkan ranah waktu yang terbaik. Ilustrasi dari proses *frame blocking* dapat dilihat pada Gambar 1.



Gambar 1 Ilustrasi *frame blocking*[10].

2.2.3. *Windowing*

Windowing merupakan proses penghalusan *spectrum*. *Windowing* memiliki tujuan untuk mengurangi efek diskontinue pada ujung-ujung *frame*[9]. *Windowing* dilakukan karena dalam pemrosesan sinyal, sinyal yang nyata memiliki batas pada waktunya[9]. Salah satu jenis *windowing* yang biasa digunakan adalah *hamming windowing*.

Fungsi *hamming windowing* digunakan seperti jendela dengan melihat blok berikutnya dalam proses ektraksi fitur dan memadukan semua garis frekuensi terdekat. Fungsi *hamming windowing* dapat dilihat pada Persamaan (2) sebagai berikut[10]:

$$0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), \wedge\ 0 \le n < N-1 \quad (2)$$

Setelah itu dikalikan dengan hasil persamaan *hamming windowing* dengan *input* yang ditetapkan menggunakan rumus di Persamaan (3) yaitu [10]:

$$Y(n) = x_1(n) \times w(n), 0 \le n < N-1 \quad (3)$$

Dimana N merupakan Banyaknya sampel tiap *frame*, Y(n) merupakan Sinyal *Output*, y(n) merupakan Sinyal *Input* dan w(n) merupakan Jendela *Hamming*.

2.2.4. *Fast Fourier Transform* (FFT)

Fast Fourier Transform (FFT) merupakan metode yang melakukan proses perubahan dari *time domain* menjadi frekuensi *domain*. Langkah ini merubah setiap *frame* N sampel dari domain waktu menjadi frekuensi *domain*. FFT berfungsi untuk merespon gelombang saluran suara dalam *domain* waktu dan mengubah konvolusi getaran celah suara[11]. Persamaan FFT dapat dilihat pada Persamaan (4) sebagai berikut [10] :

$$T(k) = \sum_{n=0}^{N-1} X(n)cos\left(\frac{2\pi kn}{N}\right) - \sum_{n=0}^{N-1} X(n)sin\left(\frac{2\pi kn}{N}\right) \quad (4)$$

Dimana T(k) merupakan Hasil FFT ke-k, X(n) merupakan Hasil perhitungan *windowing* ke-n, n merupakan Nomor urut sinyal dan k merupakan Indeks dari frekuensi (1,2, ... N).

2.2.5. *Mel Frequency Wrapping*

Mel Frequency Wrapping (MFW) merupakan *filter* berupa *filterbank* untuk mengetahui ukuran energi dari frekuensi band tertentu dalam sinyal suara[10]. Proses MFW adalah dengan mengfilter hasil FFT menjadi *mel-scale* agar dapat menyesuaikan resolusi frekuensi terhadap pendengaran manusia dan kemudian *mel-scale* tersebut menjadi sebuah *critical bank* menggunakan *filter bank*[11]. MFW dibutuhkan karena jangkauan frekuensi dalam spektrum sangat luas dan sinyal suara tidak mengikuti skala *linear*. Sehingga setelah spektrum terkomputasi, data tersebut dipetakan dalam *mel-scale* menggunakan filter segitiga yang saling tumpang tindih[9]. Persamaan dalam mencari spektrum mel dapat dilihat pada Persamaan (5) sebagai berikut [10]:

$$Y[i] = \sum_{j=1}^{G} T[j]H_i[j] \quad (5)$$

Dimana Y(i) merupakan *Mel Frequency wrapping* ke-i, G merupakan Jumlah *magnitute spectrum,* T(j)

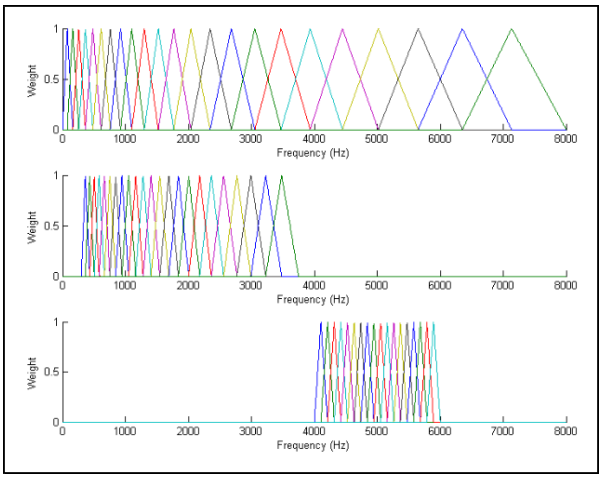
merupakan hasil FFT, $H_i(j)$, merupakan koefisien *filterbank* pada frekuensi j ($1 \leq i \leq E$) dan E merupakan jumlah *channel* dalam *filterbank*.

Sedangkan untuk persamaan yang digunakan dalam bentuk *mel* dapat dilihat pada Persamaan 6 yaitu :

$$mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \quad (6)$$

Dimana f merupakan frekuensi

Ilustrasi pada hasil dari proses MFW dapat dilihat pada Gambar 2.



Gambar 2 Contoh gambar *spectrum mel*[10].

2.2.6. *Discrete Cosine Transform* (DCT)

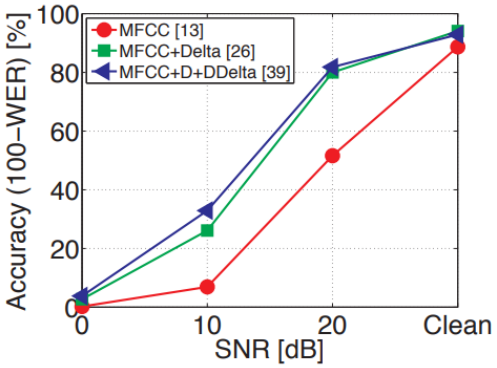
Discrete Cosine Transform (DCT) merupakan proses mendekorelasi septrum mel yang menghasilkan penyajian yang baik dari properti *spectral local*. DCT mirip sekali dengan transformasi *fourier* terdekomposisi sinyal ke gelombang cosinus[10]. DCT memiliki konsep yang sama dengan *inverse fourier transform* namun hasil DCT mendekati *principle component analysis* (PCA) yang merupakan metode statik klasik yang digunakan secara luas dalam analisis data dan kompresi. Tujuan DCT adalah untuk menghasilkan septrum mel untuk meningkatkan kualitas pengenalan. Persamaan pada DCT dapat dilihat pada Persamaan (7) sebagai berikut [10]:

$$C_m = \sum_{k=1}^{K}\left(log_{10} Y[k] cos\left[m\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right]; m = 1,2, \dots K\right) \quad (7)$$

Dimana C_m merupakan koefisien, Y(k) merupakan keluaran dari proses *filterbank* pada indeks, m merupakan banyak koefisien dan K merupakan jumlah koefisien yang diharapkan.

2.2.7. *Delta Energy* dan *Spectrum*

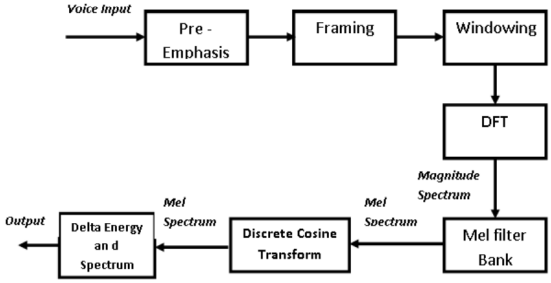
Suara dan perubahan *frame* merupakan kemiringan forman dalam transisi. Diperlukan penambahan fitur yang dapat merubah karakteristik *septral* dari waktu ke waktu[11]. Proses penambahan informasi dari fitur lereng disebut fitur delta dan proses penambahan fitur akselerasi disebut fitur akselerasi delta ganda[11]. Contoh ilustrasi pada hasil dari proses *Delta Energy* dan *Spectrum* dapat dilihat pada Gambar 3.



Gambar 3 Contoh gambar grafik MFCC menggunakan *delta spectrum*[13].

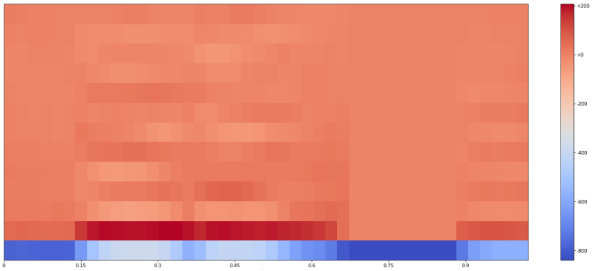
2.3. MFCC

MFCC adalah bentuk dari transformasi linear kosinus dari sinyal suara pada skala frekuensi non-linear yang diambil spectrum daya log waktu singkatnya[4]. MFCC merubah suara menjadi linear dengan frekuensi fisik nada sesuai dengan sistem pendengaran manusia. Arsitektur MFCC memiliki beberapa representasi tahap *filter* untuk melakukan ekstraksi fitur. Berikut langkah-langkah pada proses MFCC pada Gambar 4[8]. :



Gambar 4 Arsitektur MFCC dalam melakukan ekstraksi fitur.

Ilustrasi pada hasil dari proses MFW dapat dilihat pada Gambar 5.

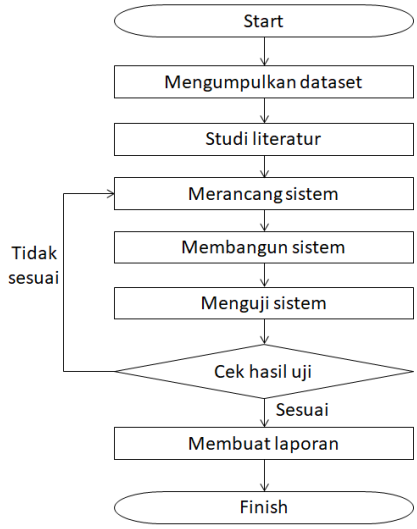


Gambar 5 Contoh gambar MFCC.

3. METODE

3.1. Diagram Alir Penelitian.

Diagram alir pada alur penelitian ini disajikan pada Gambar 6 :



Gambar 6 Diagram alir alur penelitian.

Langkah pertama yang dilakukan adalah pengumpulan dataset. Pada langkah ini dilakukan dengan mengumpulkan suara. Dataset yang digunakan terdapat 50 kata berdasarkan huruf Abjad kecuali f, q, v, x, y dan z. Tiap kata memiliki 50 suara. Cara pengumpulannya dengan mengumpulkan suara dari orang Lombok yang dimana setiap orang menyebutkan 10 kata dan 50 kali untuk setiap katanya. Kata tersebut yang disajikan pada Tabel 1 adalah sebagai berikut:

Tabel 1 Jenis kata

No	Huruf	Kata
1.	A	Adeq, Adeng, Aiq, Anjah
2.	B	Bekedek, Bareh, Berajah, Bansu
3.	C	Cakreh, Cecel
4.	D	Dahar, Doro, Dila
5.	E	Empos, Endeqman
6.	G	Gaur, Gaweq
7.	H	Harep, Hawe
8.	I	Inaq, Ima
9.	J	Jenu, Joman
10.	K	Kadi, Kadu
11.	L	Lampaq, Lekak
12.	M	Manuk, Mace
13.	N	Ndaq, Ngomeh, Nyalean
14.	O	Osok, Oat
15.	P	Padu, Polak, Pelotan, Priaq
16.	R	Rari, Rembek
17.	S	Saiq, Side, Sugih, Selapu
18.	T	Tangkong, Tindoq
19.	U	Uiq, Ulu
20.	W	Wah, Wayahan

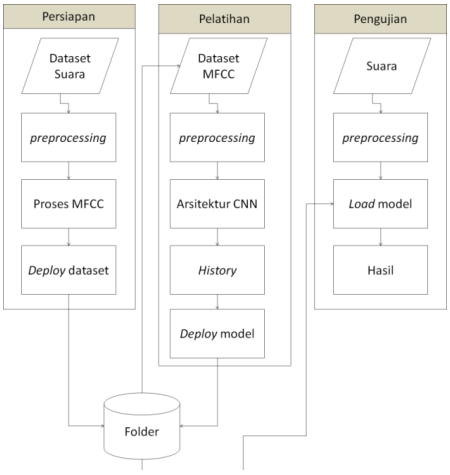
Langkah kedua adalah studi literatur. Pada langkah ini dilakukan dengan cara mencari literatur berupa jurnal, buku atau sumber lain yang berhubungan dengan penelitian. Literatur yang berhubungan dengan penelitian ini adalah literatur yang memiliki materi mengenai suara, MFCC dan CNN. Dari literatur yang dicari tersebut kemudian dipelajari mengenai teori, keunggulan dan implementasi dari metode tersebut sehingga dapat digunakan pada penelitian ini.

Langkah ketiga adalah perancangan sistem. Pada tahap ini sistem dirancang untuk menghasilkan sistem yang memiliki keterbaruan berdasarkan literatur yang dipelajari. Setelah merancang sistem maka dilakukannya pemangunan sistem berdasarkan rancangan sistem di tahap sebelumnya. Setelah pembangunan maka dilakukan uji coba pada

sistem. Jika gagal, penelitian ini akan kembali ke tahap perancangan sistem untuk mencari dan memperbaiki sistem. Jika berhasil, penelitian ini akan ke tahap terakhir yaitu pembuatan laporan berdasarkan sistem yang dibuat.

3.2. Perancangan Sistem.

Perangcangan sistem pada penelitian ini memiliki 3 tahap inti yaitu persiapan, pelatihan dan pengujian. Berikut merupakan diagram alirnya pada gambar 7:



Gambar 7 Diagram alir proses *preparing*, *training* dan *testing*.

3.2.1. *Prepraring*.

Pada tahap ini merupakan tahap untuk mempersiapkan suara dataset. Hasil dari tahap ini adalah mendapatkan *mapping*, label, MFCC dan *filenames*. *Mapping* merupakan list dari jenis kata pada dataset suara. Label merupakan list pada suara yang dimasukan dan dikelompokan dalam bentuk angka berdasarkan *mapping*. Pada penelitian ini terdapat 50 *map* yaitu kata yang digunakan pada dataset. Maka untuk pelabelan, *adeng* dilabelkan dengan 0, *adeq* dilabelkan dengan 1 dan seterusnya. *Filenames* merupakan nama *file* audio yang dipersiapkan.

MFCC merupakan hasil dari ekstraksi suara menjadi MFCC berdasarkan label. Proses MFCC ini menggunakan *library librosa* pada *python* dan diproses langsung oleh *librosa*. Parameter yang digunakan adalah sinyal, *sample rate*, jumlah koefisien, *hop length* dan jumlah FFT.

Sinyal pada parameter MFCC merupakan sinyal yang berasal dari suara masukan yang sudah diseleksi. Proses seleksinya adalah suara tersebut diseleksi apakah sampel tersebut lebih banyak dari 22050 yang dimana 22050 sampel merupakan sampel yang ada pada satu detik suara. Jika lebih sedikit maka suara tidak dibuang. Jika sama atau lebih banyak maka sampel pada suara tersebut dirubah menjadi 22050 sampel. Tujuannya adalah agar ukuran signal suara sama semua dan memudahkan untuk dilatih pada tahap berikutnya. *Sample rate* merupakan *sample rate* yang berasal dari suara yang diinput. Jumlah koefisien merupakan total koefisien yang dibutuhkan untuk diekstrak. Jumlah koefisien yang digunakan adalah 13. *Hop length* merupakan jumlah *frame* yang dibutuhkan untuk menghitung MFCC. *Hop length* yang digunakan adalah 512. Jumlah FFT merupaka jumlah *window* yang dibutuhkan untuk FFT. Jumlah FFT yang digunakan adalah 2048.

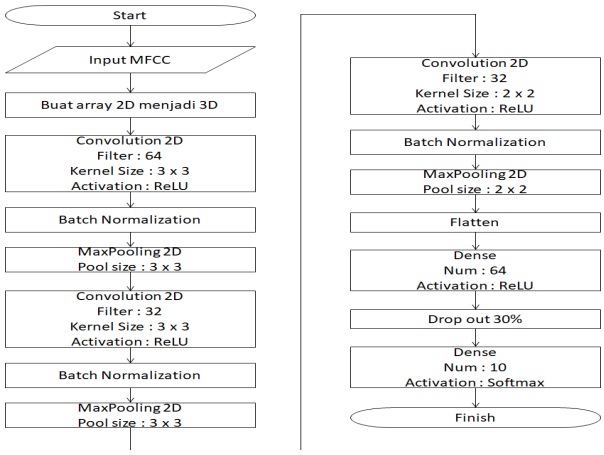
Ketika sudah diproses, hasil dari proses tersebut disimpan menjadi *file* json. *File* json tersebut terdiri dari *mapping*, label, MFCC dan *filenames*. *File* tersebut merupakan dataset untuk dilatih pada tahap selanjutnya.

3.2.2. *Training*.

Pada tahap ini merupakan tahap untuk melatih dataset yang sudah disiapkan dari tahap sebelumnya. Dataset yang sudah disiapkan menjadi MFCC akan digunakan untuk pelatihan sistem dan hasil pelatihan tersebut akan digunakan pada tahap berikutnya. Proses pelatihan menggunakan *framework tensorflow* dan *keras* sehingga dapat memudahkan dalam membuat pelatihan pada dataset. Jumlah data yang digunakan sesuai dengan jumlah MFCC dan label pada file json yang sudah diproses pada tahap persiapan yaitu 2500 MFCC dan 50 label.

Pada tahap *preprocessing* yang dilakukan adalah mempersiapkan dataset yang akan dilatih. Dari dataset tersebut diambil MFCC sebagai *input* yang disimbolkan dengan *x* dan label sebagai target *output* yang disimbolkan dengan *y*. Pada tiap *x* dan *y* dibagi menjadi 3 yaitu *x_train* dan *y_train* yang akan digunakan untuk pelatihan, *x_validation* dan *y_validation* yang digunakan untuk validasi dari hasil pelatihan, dan *x_test* dan *y_test* untuk menguji dari hasil validasi. Pada matrik MFCC dirubah dari 2D menjadi 3D yang terdiri dari *segments*, jumlah koefisien, dan channel berjumlah 1. Tujuannya adalah agar MFCC dapat diproses pada CNN.

Pada arsitektur CNN yang digunakan untuk training digambarkan dengan diagram alir seperti yang disajikan pada Gambar 8 adalah sebagai berikut :



Gambar 8 Diagram alir arsitektur CNN

Pada diagram alir tersebut memiliki penjelasan seperti yang disajikan pada Tabel 2 berikut :

Tabel 2 Penjelasan arsitektur CNN

Layer (type)	*Output Shape*	*Param*
Conv2d (Conv 2D)	(None, 42, 11, 64)	640
batch_normalization (BatchNormalization)	(None, 42, 11, 64)	256
max_pooling2d (MaxPooling2D)	(None, 21, 6, 64)	0
conv2d_1 (Conv2D)	(None, 19, 4, 32)	18464
batch_normalization_1 (BatchNormalization)	(None, 19, 4, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 10, 2, 32)	0
conv2d_2 (Conv2D)	(None, 9, 1, 32)	4128
batch_normalization_2 (BatchNormalization)	(None, 9, 1, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 5, 1, 32)	0
flatten (Flatten)	(None, 160)	0
dense (Dense)	(None, 64)	10304
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 10)	650

Pada tabel 3.3 terdiri dari *layer (type)*, *Output Shape*, *Param*. *layer (type)* merupakan jenis lapisan yang digunakan pada arsitektur CNN. *Output Shape* merupakan bentuk keluaran yang dihasilkan pada tiap lapisan dan akan menjadi masukan di lapisan berikutnya. Pada *output shape* terdiri dari *batch_size*, *height*, *width*, dan *depth*. *Param* merupakan parameter yang dilatih pada tiap lapisan.

Dari arsitektur CNN pada gambar 3.3 terdapat 13 *layer* CNN. *Layer* tersebut terdiri dari 3 konvolusi, 3 BN, 3 *Max pooling*, Flatten, 2 dense dan *dropout*. Data yang dimasukan untuk dilatih berbentuh matriks 3 dimensi yang terdiri dari *segments*, jumlah koefisien berjumlah 13, dan channel berjumlah 1. Kelas yang dihasilkan ada 50 berdasarkan jumlah kata pada dataset.

Pada lapisan konvolusi merupakan tempat terjadinya proses konvolusi pada matriks MFCC. Tujuan dari *convolution layer* adalah untuk melakukan *filter* terhadap matrik yang dimasukan. Proses pada konvolusi adalah dengan menghitung serangkaian aktivasi pada matriks MFCC yang tertutupi matriks kernel kemudian matriks kernel tersebut bergeser dan menghitung dan seterusnya. Untuk mempertahankan ukuran matriks dibutuhkan *zero padding*. Pada penelitian ini, kernel yang digunakan berukuran 3x3. Untuk meningkatkan kecepatan dalam proses training maka digunakan lapisan BN. Hasil dari lapisan konvolusi ini merupakan masukan untuk lapisan *pooling*.

Pada lapisan *max pooling* merupakan proses untuk mengurangi ukuran dari input sehingga proses feature map jadi lebih cepat. Pada penelitian ini menggunakan matriks sebesar 3x3 yang dimana dari matrik tersebut dicari nilai terbesarnya yang disebut dengan *max pooling*. *Stride* yang digunakan adalah 2 yang dimana matriks 3x3 tersebut akan bergeser sebanyak 2 indeks.

Pada lapisan *flatten* merupakan lapisan untuk memproses matrix hasil *max pooling* menjadi *vektor* atau menjadi 1 dimensi. Hasil dari lapisan *flatten* akan dipindan ke lapisan *dense*. Pada lapisan *dense* digunakan untuk menambahan lapisan dengan cara merubah *vektor* dengan cara melakukan perkalian pada *vektor* dan dapat menerapkan operasi rotasi, penskalaan, dan terjemahan pada vektor. Untuk menghindari terjadinya *overfitting*, maka digunakannya *dropout layer*. Pada lapisan ini akan menghapus neuron yang akan dimatikan secara acak dan bersifat sementara.

Pada lapisan *softmax* digunakan untuk menghitung distribusi probabilitas dari kejadian N dari berbagai kejadian. Pada lapisan ini akan menghitung probabilitas masing-masing kelas tujuan dari semua kelas yang ada dan kemudian menentukan kelas tujuan dari *input*. Pada penelitian

ini terdapat 50 kelas yang berarti pada lapisan ini akan menentukan kelas dari setiap masukan berdasarkan 50 kelas.

Pada saat melakukan pelatihan dibutuhkan parameter yaitu model, epochs, *batch size*, *earlystop callback*, X_train, y_train, X_validation, y_validation. Model merupakan arsitektur CNN yang sudah dibuat dan digunakan sebagai model dalam melakukan pelatihan. Epochs merupakan jumlah *network* untuk melihat dataset pada saat melatih dataset. Jumlah epochs yang digunakan adalah 51. *Batch size* merupakan jumlah sampel yang diproses sebelum model di-update. Jumlah *batch size* yang digunakan adalah 32. *earlystop callback* merupakan metode untuk menentukan periode pelatihan dan menghentikan pelatihan setelah kinerja model berhenti meningkat pada set data validasi yang tertunda. Parameter yang digunakan adalah min-delta dan patience. min-delta digunakan untuk mempertimbangkan peningkatan pada setiap perubahan. min-delta yang digunakan adalah 0.001. *Patience* merupakan jumlah epochs yang menunggu sebelum dihentikan oleh earlystop callback. Jumlah *patience* yang dibutuhkan adalah 5.

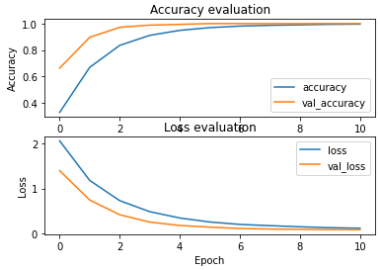
3.2.3. *Testing.*

Pada tahap ini merupakan tahap untuk menguji coba apakah sistem dapat memprediksi suara dan merubah menjadi teks dengan benar. Masukan yang digunakan adalah suara yang langsung direkam pada saat itu. Suara tersebut kemudian dijadikan MFCC. Hasil dari MFCC tersebut kemudian diprediksi menggunakan model pada tahap *training*. Setelah melewati maka akan muncul kata yang sesuai dengan suara masukan berdasarkan hasil *training*.

3.3. Teknik dan Skenario Pengujian Sistem.

Skenario pengujian sistem dilakukan untuk mengetahui akurasi terhadap jumlah dataset yang digunakan. Hasil uji yang dilakukan dikatakan baik jika akurasi diatas 70.00%. Rasio data latih dan data uji pada penelitian ini sebesar 7:3. *Learning rate* yang digunakan 0.0001. Pengujian sistem dilakukan menggunakan dua metode yaitu pelatihan CNN.

Pada pelatihan CNN menggunakan fit model untuk mendapatkan nilai akurasi dan nilai loss pada hasil latihan data. Pada fit model menggunakan x_train dan y_train sebagai data latih, epoch sejumlah 56, batch_size sejumlah 32, patience sejumlah 5, dan x_validation dan y_validation sebagai validasi. Contoh grafik pada pelatihan CNN dapat dilihat pada Gambar 9.



Gambar 9 Contoh grafik akurasi dan grafik *loss*.

Hasil dari pelatihan fit model akan menampilkan 2 grafik seperti pada contoh gambar 3.4 dan menampilkan nilai akurasi dan nilai loss. Kedua grafik tersebut terdiri dari grafik evaluasi akurasi dan grafik evaluasi loss. Pada kedua grafik tersebut terdapat y sebagai akurasi untuk grafik akurasi dan sebagai loss untuk grafik loss dan x sebagai *epoch*. Kemudian terdapat 2 garis yang dimana garis pertama menandakan pergerakan data training dan garis kedua menandakan pergerakan nilai akurasi untuk grafik akurasi dan nilai *loss* untuk grafik *loss*. Nilai akurasi merupakan nilai yang menampilkan tingkat keberhasilan model yang dibuat. Sedangkan nilai loss merupakan suatu ukuran *error* yang dibuat oleh jaringan yang bertujuan untuk meminimalisirnya.

4. HASIL DAN PEMBAHASAN

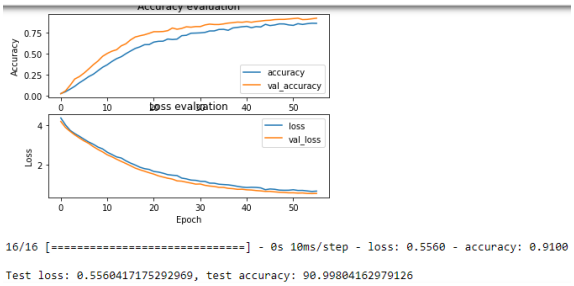
4.1. Mekanisme Penelitian.

Hasil yang didapatkan dalam pengujian penelitian ini adalah dengan mendapatkan akurasi yang dihasilkan setelah melakukan pelatihan pada dataset suara. Tahapan yang pertama adalah dengan mengekstraksi fitur pada dataset menggunakan MFCC. Tahap kedua yaitu melatih hasil dari ekstraksi fitur menggunakan model CNN. Tahapan ketiga adalah dengan mencoba hasil dari pelatihan menggunakan model CNN. Tahapan terakhir adalah mencoba *speech to text* dengan mencoba mengatakan 50 kata dan perkata sebanyak 3 kali.

4.2. Hasil Pelatihan.

Hasil dari pelatihan merupakan hasil dari pelatihan menggunakan klasifikasi CNN. Fungsi yang digunakan pada pelatihan CNN adalah fungsi yang terdapat pada *library* tensorflow pada Python. Pada hasil pelatihan CNN terdapat 2 grafik yaitu *accuracy evaluation* dan *loss evaluation*. Pada *accuracy evaluation*, garis akurasi dan nilai akurasi meningkat bersamaan dan berdekatan. Untuk *loss evaluation*, garis *loss* dan nilai *loss* menurun bersamaan dan berdekatan. Kecepatan dalam pelatihan yang dilakukan pada klasifikasi CNN adalah 10s/step. Dari

pelatihan tersebut menghasilkan hasil *test loss* adalah 0.556 dan *test accuracy* adalah 90.998. Sehingga, akurasi pada hasil pelatihan CNN lebih besar dari 70% yaitu 90% dan dapat dikatakan baik. Untuk hasil dari pelatihan dapat dilihat dari Gambar 10:



Gambar 10 Hasil dari pengujian klasifikasi CNN.

4.3. Hasil Pengujian.

Pengujian yang dilakukan adalah dengan mencoba merubah suara menjadi teks. Uji coba tersebut dilakukan dengan mengatakan 50 kata sesuai dengan dataset dan dilakukan sebanyak 3 kali per kata. Contoh pengujian dataset menggunakan sampel dapat diliat pada gambar 11:

```
durasi = librosa.get_duration(filename='sample/Bareh (1).wav')
print(durasi)
keyword1 = kss.predict("sample/Bareh (1).wav")
print(keyword1)
```

```
1.1519954648526076
Bareh
```

Gambar 11 Hasil dari pengujian klasifikasi CNN.

Dari gambar 11, fungsi durasi digunakan untuk mengetahui panjang durasi dari sampel *bareh* yang digunakan pada pengujian. Sedangkan fungsi *keyword1* digunakan untuk menguji dataset menggunakan sampel kata *bareh*.Untuk hasil dari pengujian dapat dilihat dari Tabel 3 berikut:

Tabel 3 Hasil pengujian kata pada dataset.

Kata	Benar	Salah
Adeq	2	1
Adeng	3	0
Aiq	3	0
Anjah	3	0
Bekedek	3	0
Bareh	1	2
Berajah	3	0
Bansu	3	0
Cakreh	3	0
Cecel	3	0
Dahar	2	1
Doro	3	0
Dila	3	0
Empos	3	0
Endeqman	3	0
Gaur	3	0
Gaweq	3	0
Harep	2	1
Hawe	3	0
Inaq	2	1
Ima	3	0
Jenu	3	0
Joman	3	0
Kadi	3	0
Kadu	3	0
Lampaq	3	0
Lekak	2	1
Manuk	3	0
Mace	3	0
Ndaq	3	0
Ngomeh	3	0
Nyalean	3	0
Osok	3	0
Oat	3	0
Padu	2	1
Polak	3	0
Pelotan	3	0
Priaq	3	0
Rari	3	0
Kata	Benar	Salah
Rembek	3	0
Saiq	3	0
Side	3	0
Sugih	3	0
Selapu	3	0
Tangkong	3	0
Tindoq	3	0
Uiq	3	0
Ulu	3	0
Wah	3	0
Wayahan	3	0

Dari tabel Tabel 4.2 dapat terlihat bahwa terdapat 43 kata yang benar semua, 6 kata yang benar 2, 1 kata yang benar 1 dan tidak ada kata yang salah semua. Sehingga memiliki persentase 86% benar semua, 12% benar 2, dan 2% benar 1, dan 0% salah semua.

5. KESIMPULAN DAN SARAN

Kesimpulan yang didapatkan berdasarkan hasil dari penelitian ini dapat disimpulkan dalam berikut:

1. Pembuatan *speech to text conversion* dapat dilakukan menggunakan ekstraksi fitur MFCC dan klasifikasi CNN dengan hasil 43 kata yang benar semua, 6 kata yang benar 2, 1 kata yang benar 1 dan tidak ada kata yang salah semua. Sehingga memiliki

persentase 86% benar semua, 12% benar 2, dan 2% benar 1, dan 0% salah semua.

2. Akurasi yang didapatkan dari hasil percobaan adalah 90% dengan menggunakan klasifikasi model CNN dengan arsitektur *learning rate* menggunakan 0.0001, *batch size* 32, *patience* 5 dan *epoch* 56 Arsitektur yang digunakan untuk pelatihan CNN terdiri dari 3 lapisan konvolusi, 3 lapisan BN, 3 *max pooling*, *flatten*, 2 *dense* dan *dropout*.

Saran yang dapat disampaikan dari hasil penelitian ini jika penelitian ini ingin dikembangkan adalah sebagai berikut:

1. Jumlah kata yang digunakan pada dataset lebih banyak dan lebih bervariasi.
2. Peningkatan penggunaan data pada setiap kelas akan berpotensi meningkatkan akurasi dari algoritma yang dirancang
3. Hasil luaran yang dihasilkan dapat lebih banyak atau bahkan bisa menjadi satu kalimat.

UCAPAN TERIMA KASIH

Sebagai penulis penelitian ini mengungkapkan terima kasih yang tulus kepada dosen pembimbing yang ingin membimbing tugas akhir saya, rekan-rekan yang mau membantu saya dalam pengerjaan skripsi, dan keluarga yang selalu *support* kepada penulis.

DAFTAR PUSTAKA

[1] E. Widiyanto, S. N. Endah, and S. Adhy, "Aplikasi Speech To Text Berbahasa Indonesia Menggunakan Mel Frequency Cepstral Coefficients Dan Hidden Markov Model (Hmm)," *Pros. Semin. Nas. Ilmu Komput. Undip*, pp. 39–44, 2014.

[2] Lalu Erwan Husnan, "EJAAN BAHASA SASAK SASAK LANGUAGE SPELLING Lalu Erwan Husnan," no. 2005, pp. 27–35, 2012.

[3] S. Wilian, "Tingkat Tutur dalam Bahasa Sasak dan Bahasa Jawa," *Wacana, J. Humanit. Indones.*, vol. 8, no. 1, p. 32, 2006, doi: 10.17510/wjhi.v8i1.245.

[4] D. Lionel, R. Adipranata, and E. Setyati, "Klasifikasi Genre Musik Menggunakan Metode Deep Learning Convolutional Neural Network dan Mel- Spektrogram," *J. Infra Petra*, vol. 7, no. 1, pp. 51–55, 2019, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8044.

[5] V. Bansal, G. Pahwa, and N. Kannan, "Cough classification for COVID-19 based on audio mfcc features using convolutional neural networks," *2020 IEEE Int. Conf. Comput. Power Commun. Technol. GUCON 2020*, pp. 604–608, 2020, doi: 10.1109/GUCON48875.2020.9231094.

[6] S. Tripathi, A. Kumar, A. Ramesh, C. Singh, and P. Yenigalla, "Deep learning based emotion recognition system using speech features and transcriptions," *arXiv*, vol. 1, pp. 1–12, 2019, [Online]. Available: https://arxiv.org/abs/1906.05681.

[7] Z. I. R. and H. Z. A. 1 A. Zabidi1 , I. M. Yassin1,* , H. A. Hassan2 , N. Ismail1 , M. M. A. M. Ham ah zah1, "Detection of Asphyxia in Infants Using Deep Learning Ction of Asphyxia in Infants Using Deep Learning Convolutional Neural Network (Cnn) Trained on Mel Frequency Olutional Neural Network (Cnn) Trained on Mel Frequency Cepstrum Coefficient (Mfcc) Features," *Aust. ranger Bull.*, vol. 4, no. 1, pp. 768–778, 2017, doi: 10.4314/jfas.v9i3s.59.

[8] M. Alhlffee, "MFCc-based feature extraction model for long time period emotion speech using cnn," *Rev. d'Intelligence Artif.*, vol. 34, no. 2, pp. 117–123, 2020, doi: 10.18280/ria.340201.

[9] T. Chamidy, "Metode Mel Frequency Cepstral Coeffisients (MFCC) Pada klasifikasi Hidden Markov Model (HMM) Untuk Kata Arabic pada Penutur Indonesia," *Matics*, vol. 8, no. 1, pp. 36–39, 2016, doi: 10.18860/mat.v8i1.3482.

[10] H. Heriyanto, S. Hartati, and A. E. Putra, "Ekstraksi Ciri Mel Frequency Cepstral Coefficient (Mfcc) Dan Rerata Coefficient Untuk Pengecekan Bacaan Al-Qur'an," *Telematika*, vol. 15, no. 2, p. 99, 2018, doi: 10.31315/telematika.v15i2.3123.

[11] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using Mel Frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques," *CONIELECOMP 2012 - 22nd Int. Conf. Electron. Commun. Comput.*, vol. 2012-Febru, no. February, pp. 248–251, 2012, doi: 10.1109/CONIELECOMP.2012.6189918.

[12] S. R. Subramanya, A. Youssef, B. Narahari, R. Simha, and T. George, "Audio Data Indexing Using Discrete Cosine Transform .," no. January, 2003.

[13] C. K. and R. S. Kshitiz Kumar, "DELTA-SPECTRAL CEPSTRAL COEFFICIENTS FOR ROBUST SPEECH RECOGNITION Carnegie Mellon University , Pittsburgh , PA 15213 Email : { kshitizk , chanwook , rms }@ cs . cmu . edu," *Science (80-.).*, pp. 1–4, 2011.

[14] H. Mulyana, "Klasifikasi Citra Pornografi Dengan Metode Convolutional Neural Network Pada Perangkat Smartphone Berbasis Android," *Univ.*

Mataram, 2020, [Online]. Available: http://begawe.unram.ac.id/index.php/ta/article/view/10.

[15] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv Prepr.*, vol. 2015-Decem, no. December, pp. 1–11, 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

[16] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding Batch Normalization - Google 搜索," *arXiv Prepr.*, vol. 4, no. NeurIPS, pp. 1–24, 2018, [Online]. Available: https://www.google.ca/search?q=Understanding+Batch+Normalization&oq=Understanding+Batch+Normalization&aqs=chrome..69i57j69i61j69i60j0.588j0j7&client=ubuntu&sourceid=chrome&ie=UTF-8.

[17] Y. Achmad, R. C. Wihandika, and C. Dewi, "Klasifikasi emosi berdasarkan ciri wajah wenggunakan convolutional neural network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 11, pp. 10595–10604, 2019, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6732.

[18] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. April 2018, pp. 2607–2612, 2016, doi: 10.1109/ITSC.2016.7795975.

[19] B. Mele and G. Altarelli, "Lepton spectra as a measure of b quark polarization at LEP," *Phys. Lett. B*, vol. 299, no. 3–4, pp. 345–350, 1993, doi: 10.1016/0370-2693(93)90272-J.

[20] W. You, C. Shen, X. Guo, X. Jiang, J. Shi, and Z. Zhu, "A hybrid technique based on convolutional neural network and support vector regression for intelligent diagnosis of rotating machinery," *Adv. Mech. Eng.*, vol. 9, no. 6, pp. 1–17, 2017, doi: 10.1177/1687814017704146.