TESIS

ANALISIS EKSTRAKSI FITUR AUDIO PADA *CONVOLUTIONAL NEURAL NETWORK* UNTUK PENGENALAN AKSEN

Disusun oleh:

Nama	**: Dwi Sari Widyowaty**
NIM	**: 19.51.1234**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2021

TESIS

ANALISIS EKSTRAKSI FITUR AUDIO PADA *CONVOLUTIONAL NEURAL NETWORK* UNTUK PENGENALAN AKSEN

AUDIO FEATURE EXTRACTION ANALYSIS ON CONVOLUTIONAL NEURAL NETWORK FOR ACCENT RECOGNITION

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Dwi Sari Widyowaty**
NIM	**: 19.51.1234**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2021

HALAMAN PENGESAHAN

ANALISIS EKSTRAKSI FITUR AUDIO
PADA *CONVOLUTIONAL NEURAL NETWORK*
UNTUK PENGENALAN AKSEN

AUDIO FEATURE EXTRACTION ANALYSIS
ON CONVOLUTIONAL NEURAL NETWORK
FOR ACCENT RECOGNITION

Dipersiapkan dan Disusun oleh

Dwi Sari Widyowaty

19.51.1234

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 5 Oktober 2021

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Oktober 2021
Rektor

Prof. Dr. M. Suyanto, M.M
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS EKSTRAKSI FITUR AUDIO
PADA *CONVOLUTIONAL NEURAL NETWORK*
UNTUK PENGENALAN AKSEN

AUDIO FEATURE EXTRACTION ANALYSIS
ON CONVOLUTIONAL NEURAL NETWORK
FOR ACCENT RECOGNITION

Dipersiapkan dan Disusun oleh

Dwi Sari Widyowaty

19.51.1234

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 5 Oktober 2021

Pembimbing Utama

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Anggota Tim Penguji

Prof. Dr. Kusrini, M.Kom
NIK. 190302106

Pembimbing Pendamping

Alva Hendi Muhammad, S.T., M.Eng., Phd
NIK. 190302493

Hanif Al Fatta, M.Kom
NIK. 190302096

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Oktober 2021
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Dwi Sari Widyowaty
NIM : 19.51.1234
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Ekstraksi Fitur Audio pada *Convolutional Neural Network* untuk Pengenalan Aksen

Dosen Pembimbing Utama : Andi Sunyoto
Dosen Pembimbing Pendamping : Hanif Al Fatta

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 5 Oktober 2021
Yang Menyatakan,

Dwi Sari Widyowaty

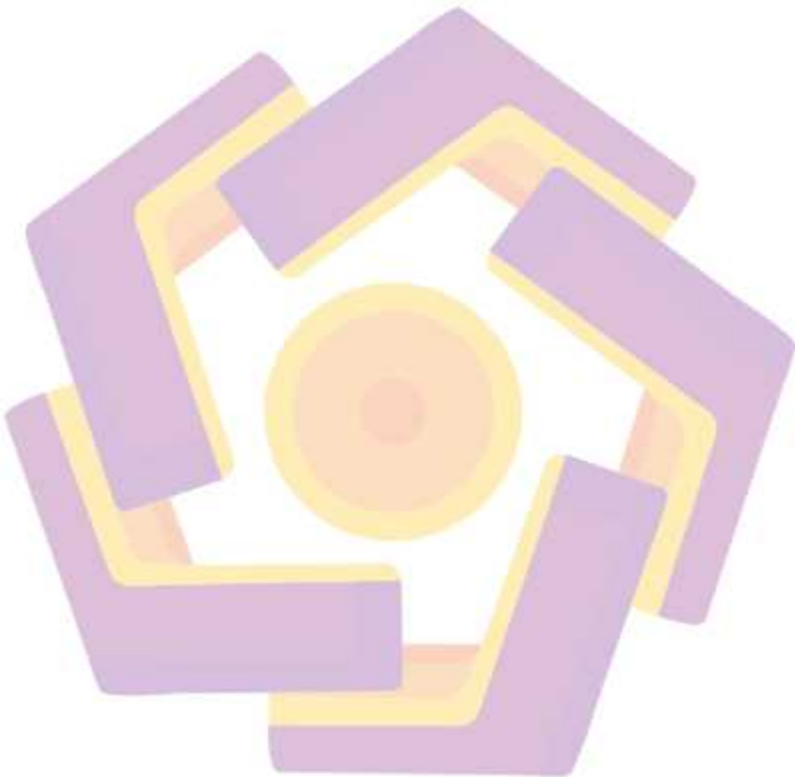
v

HALAMAN PERSEMBAHAN

Tesis ini Saya persembahkan untuk Suami dan Anak saya tercinta "Fadhilatu Nuruzzahra W". Terima kasih atas segala do'a, semangat, dukungan, perhatian, motivasi serta kasih sayang yang diberikan selama ini. Semoga kelak tesis ini dapat menjadi *support system* bagi anak-anak ku untuk terus semangat dalam meraih pendidikan setinggi-tingginya.

HALAMAN MOTTO

"Tidak ada halangan untuk meraih mimpi, percaya lah jika kamu yakin, kamu pasti bisa"



KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT karena atas segala karunia dan ridho-Nya, sehingga tesis yang berjudul "Analisis Ekstraksi Fitur Audio pada *Convolutional Neural Network* untuk Pengenalan Aksen" dapat diselesaikan dengan tepat waktu.

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister Komputer pada program studi Magister Teknik Informatika Universitas Amikom Yogyakarta.

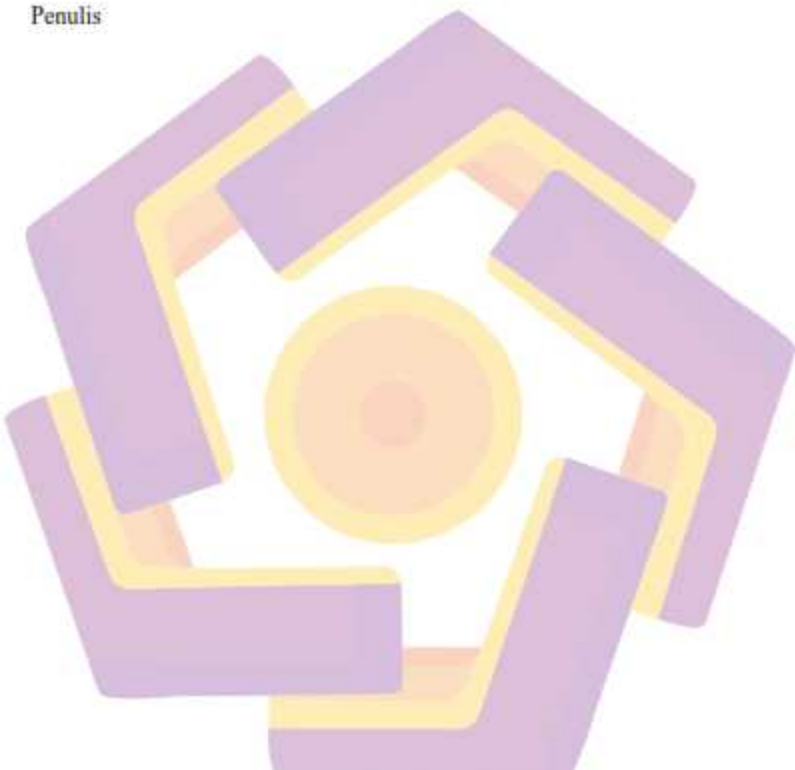
Penyelesaian tesis yang sangat berharga ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Pada kesempatan ini, penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Kedua Orang Tua, Suami, dan Anak tercinta yang senantiasa memberikan do'a, semangat, dan dukungan kepada penulis agar senantiasa semangat dalam menuntut ilmu.
2. Bapak Dr. Andi Sunyoto, M.Kom. selaku pembimbing utama yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
3. Bapak Hanif Al Fatta, M.Kom. selaku pembimbing pendamping yang telah membimbing, membantu, dan memotivasi dalam penulisan tesis ini sehingga dapat terselesaikan dengan baik.
4. Dosen Penguji yang telah memberikan saran yang baik demi kemajuan tesis ini.

5. Direktur Program Pascasarjana, jajarannya, staf dan rekan-rekan Magister Teknik Informatika Universitas Amikom Yogyakarta.

Yogyakarta, 5 Oktober 2021

Penulis



DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Aksen merupakan pelafalan yang khas dari seorang pembicara, perbedaan aksen dipengaruhi oleh lingkungan, budaya, dan asal negara. Pengenalan aksen merupakan bagian dari *Automatic Speech Recognition (ASR)*. Trend ASR telah banyak diimplementasikan pada berbagai alat elektronik, contohnya *smartphone, laptop* atau *speaker* dengan fitur *voice assistant*. Alat elektronik dengan menggunakan fitur suara harus bekerja keras agar dapat melakukan pekerjaan sesuai dengan yang diharapkan pengguna. Tujuan penelitian aksen yaitu untuk meningkatkan performa system pengenalan suara agar *voice assistant* menjadi lebih canggih. Pada penelitian ini, penulis mengklasifikasikan aksen dari 6 negara yaitu *Arabic*, *English*, *French*, *Korean*, *Mandarin* dan *Spanish*. Dataset yang digunakan berjumlah 1023 rekaman suara dengan format MP3, masing-masing pembicara mengucapkan *script* dalam Bahasa Inggris. Dataset terdiri dari 102 *Arabic*, 579 *English*, 63 *French*, 52 *Korean*, 65 *Mandarin*, dan 162 *Spanish*. Fitur audio yang digunakan yaitu *Zero Crossing Rate* (ZCR) dan *Energy*, dataset yang berupa audio akan diekstrak menjadi array angka berukuran 1 dimensi dan kemudian diolah menjadi 2 dimensi agar dapat diklasifikasikan menggunakan *Convolutional Neural Network* (CNN). Penggunaan Fitur ZCR mendapatkan akurasi 62 % pada durasi 15 detik, dan fitur *Energy* mendapatkan akurasi 60 % pada durasi 2 dan 6 detik. Akurasi yang telah dihasilkan dapat lebih tinggi daripada penelitian sebelumnya yaitu di atas 53 %.

Kata kunci: Pengenalan Aksen, Fitur Audio, *Zero Crossing Rate*, *Energy*, CNN

ABSTRACT

An accent is a typical pronunciation of a speaker, accent differences are influenced by environment, culture, and country of origin. Accent recognition is part of Automatic Speech Recognition (ASR). The ASR trend has been widely implemented on various electronic devices, for example, smartphones, laptops or speakers with voice assistant features. Electronic devices using the voice feature must work hard to do the job as expected by the user. The purpose of accent research is to improve the performance of the voice recognition system, so that voice assistants become more sophisticated. In this study, the author tries to classify accents from 6 countries: Arabic, English, French, Korean, Mandarin, and Spanish. The dataset used is 1023 MP3 voice recordings, each speaker recorded with the same script in English. The dataset consists of 102 Arabic, 579 English, 63 French, 52 Korean, 65 Mandarin, and 162 Spanish. The audio features used are Zero Crossing Rate (ZCR) and Energy. The audio dataset will be extracted 1-dimensional array and then processed into a 2-dimensional array to be classified using Convolutional Neural Network. Feature ZCR gets an accuracy of 62 % on 15 seconds, and feature energy gets an accuracy of 60 % on 2 and 6 seconds. The result produced can be higher than previous research, which is above 53 %.

Keyword: Accent Recognition, audio features, Zero Crossing Rate, Energy, CNN

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Aksen dalam Bahasa Indonesia dapat dikatakan logat atau pelafalan yang khas dari seorang pembicara, aksen dapat dipengaruhi oleh lingkungan, budaya, dan asal negara (Barkana & Patel, 2020). Aksen dari setiap negara berbeda-beda, misalnya Aksen Amerika dengan Aksen Inggris, Aksen Korea dengan Aksen Mandarin, dan sebagainya. Pengenalan aksen merupakan bagian dari *Automatic Speech Recognition (ASR).* Trend ASR telah banyak diimplementasikan pada berbagai software, penggunaan ASR dapat bermanfaat untuk membantu atau mengelola pekerjaan manusia. Untuk mendukung teknologi ASR, perlu dilakukan penelitian pengenalan aksen karena pengenalan aksen dapat meningkatkan performa dari Sistem Pengenalan Suara (Honnavalli & Shylaja, 2019).

Salah satu contoh aplikasi ASR adalah *voice assistant*. Sekarang ini, penggunaan fitur *voice assistant* telah banyak diterapkan pada alat-alat elektronik seperti *smartphone, laptop, speaker*, bahkan *lift* dengan teknologi suara. Ketika seorang pengguna alat elektronik menggunakan fitur suara untuk memberikan perintah, terkadang alat elektronik tersebut harus bekerja keras agar dapat melakukan sesuai dengan perintah. Teknologi yang baik adalah teknologi yang dapat menyesuaikan pengguna, jika pengguna tidak terbiasa ber-Bahasa inggris atau memiliki *pronouncation* yang tidak baik, teknologi yang baik tidak akan menyalahkan pengguna, melainkan teknologi yang baik akan berusaha untuk

mengenali maksud dari pengguna tersebut, agar dapat menjalankan perintah seperti yang diharapkan. Kenyataannya, teknologi *voice assistant* tidak se-sempurna yang diharapkan, terkadang *voice assistant* masih saja tidak mengerti maksud dari pengguna. Hal ini dapat dimaklumi karena faktor aksen begitu berpengaruh agar teknologi *voice assistant* dapat berjalan seperti apa yang diharapkan.

Penggunaan *voice assistant* tentu sangat mempermudah keperluan manusia, misalnya *speaker* dengan fitur *voice assistant* dapat memainkan musik sesuai dengan perintah suara. Ketika seorang pengguna me-*request* sebuah *music* dengan menyebutkan judul lagu, *voice assistant* seharusnya dapat dengan mudah memainkan judul lagu tersebut. Apabila *voice assistant* melakukan kesalahan dalam memilih *music*, dapat dipastikan performa *voice assistant* masih belum sempurna dan belum dimasukkan kemampuan mengenali aksen.

Perusahaan-perusahaan teknologi tentu telah membaca *trend* di masa yang akan datang, dimana semua alat elektronik berangsur dapat dikendalikan dengan perintah suara. Kini, penggunaan ASR mulai diterapkan dan performa system akan terus ditingkatkan agar alat elektronik dapat menjadi teman *virtual* manusia dalam kehidupan sehari-hari. Kedepan, kemampuan *virtual assistant* dapat lebih canggih, misalnya menebak latar belakang, budaya, bahasa asli/daerah, bahkan merekomendasikan sebuah makanan berdasarkan daerah asal, hal ini dapat dilakukan dengan pengenalan suara. Agar *virtual assistant* dapat lebih canggih seperti tersebut, tentu saja *virtual assistant* perlu disuntikkan kemampuan dalam mengenali aksen pengguna.

Deep learning merupakan salah satu metode yang telah diangkat dalam beberapa penelitian pengenalan aksen, diantaranya penelitian *Arabic, Italian, Japanese* dan *Korean* (4 aksen) dengan menggunakan ekstraksi fitur MFCC dan *Convolutional Neural Network*. Berdasarkan hasil penelitian tersebut, aksen Italian memberikan performa paling baik dibandingkan dengan aksen lainnya yaitu *Italian* mampu mencapai akurasi sebesar 86,9 % (Chionh et al., 2018), namun penelitian ini belum mencoba memasukkan aksen *English*. Kemudian, penelitian berikutnya membandingkan beberapa fitur menggunakan CNN untuk mendapatkan fitur mana yang memberikan performa paling baik, fitur-fitur yang digunakan meliputi MFCC, *Spectrogram, Chromagram, Spectral Centroid,* dan *Spectral Roll Off*, hasil dari penelitian ini, MFCC dan CNN memberikan performa paling baik dibandingkan fitur lainnya yaitu akurasi 48,24 % dengan segment paragraph (Singh et al., 2019), selanjutnya peneliti yang sama membandingkan performa beberapa fitur dan CNN menggunakan 5 kelas (*English*, *Spanish*, *Arabic*, *Mandarin* dan *French*) dan 3 kelas (*Arabic*, *English* dan *Mandarin*) penelitian tersebut berpendapat bahwa percobaan pada 3 kelas mampu mencapai akurasi lebih baik (akurasi dengan segment file 71,43 %) daripada 5 kelas (48,24 %), rendahnya akurasi dikarenakan tingkat keragaman pembicara yaitu tidak ada pola pasti yang dapat membedakan aksen satu pembicara dengan pembicara lainnya (Singh et al., 2020).

Tahapan pengenalan suara meliputi ekstraksi fitur sinyal audio menjadi array angka, kemudian proses klasifikasi suara. Fitur audio dibagi menjadi fitur audio domain frekuensi dan fitur audio domain waktu (Jondya & Iswanto, 2018). Penelitian yang telah dipaparkan sebelumnya menggunakan fitur audio domain

frekuensi yaitu MFCC, *Spectrogram*, *Chromagram*, *Spectral Centroid*, dan *Spectral Roll Off* (Chionh et al., 2018; Singh et al., 2019, 2020), sedangkan fitur audio domain waktu meliputi *Zero Crossing Rate* (ZCR) dan *Energy* belum dilakukan penelitian dalam bidang aksen tetapi pernah dilakukan penelitian di bidang analysis huruf vokal dan memberikan perfoma yang baik dalam klasifikasi huruf vokal (Barkana & Patel, 2020; Patel & Barkana, 2018). Karena ZCR dan *Energy* belum pernah dilakukan penelitian pada bidang aksen, maka tesis ini akan meneliti fitur audio ZCR dan *Energy*.

Berdasarkan latar belakang yang telah dipaparkan, Penulis membandingkan performa dari ZCR - CNN dan *Energy* – CNN terhadap 6 kelas yaitu *English*, *Spanish*, *Arabic*, *French*, *Korean* dan *Mandarin*, ke-enam kelas tersebut akan berbicara dalam Bahasa Inggris dengan mengucapkan *script* yang sama. Hasil dari penelitian ini bertujuan untuk menemukan fitur audio terbaik beserta dengan model CNN yang dapat menghasilkan performa terbaik dalam melakukan pengenalan aksen.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

1. Berapa tingkat akurasi yang didapatkan dari fitur ZCR dan klasifikasi CNN pada pengenalan aksen?
2. Berapa tingkat akurasi yang didapatkan dari fitur *Energy* dan klasifikasi CNN pada pengenalan aksen?

3. Fitur audio manakah yang terbaik untuk digabungkan dengan klasifikasi CNN pada pengenalan aksen?
4. Apakah tingkat akurasi yang diperoleh dari dataset 6 kelas dapat lebih baik jika dibandingkan dengan akurasi dari 5 kelas?

1.3. Batasan Masalah

Bagian ini memuat penjelasan tentang:

1. Penelitian ini menggunakan 6 (enam) aksen yaitu *English, Spanish, Arabic, French, Korean* dan *Mandarin*.
2. Setiap rekaman suara melafalkan kalimat dalam bahasa inggris.
3. Penelitian ini membandingkan performa 2 fitur audio, yaitu ZCR dan *Energy*.
4. Penelitian ini berfokus pada tahap ekstraksi fitur audio dan klasifikasi CNN.
5. Penelitian ini menggunakan *library* librosa untuk ekstraksi fitur dan *library* Scikit Learn serta Keras untuk CNN.
6. Hasil yang diberikan merupakan pemilihan fitur audio terbaik, model CNN yang digunakan dan tingkat akurasi yang dicapai.
7. Penelitian ini tidak sampai ke rekonstruktur audio.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menentukan ekstraksi fitur mana yang paling sesuai untuk digabungkan dengan klasifikasi CNN dalam pengenalan aksen.
2. Memperoleh tingkat akurasi dari fitur ZCR dan klasifikasi CNN.

3. Memperoleh tingkat akurasi dari fitur *Energy* dan klasifikasi CNN.
4. Sebagai salah satu syarat kelulusan dalam menempuh program Studi S2 Teknik Informatika pada Program Pascasarjana Universitas Amikom Yogyakarta.

1.5. Manfaat Penelitian

Manfaat dari penelitian yang dibahas pada Tesis ini sebagai berikut:

1. Untuk mengetahui ekstraksi fitur mana yang memberikan performa terbaik pada Metode CNN, hasil analisa yang diperoleh dapat menjadi acuan pemilihan Metode ekstraksi fitur pada aplikasi pengenalan aksen.
2. Berguna dalam menambah ilmu pengetahuan yang baru sebagai salah satu cabang ilmu pengenalan suara (speech recognition).
3. Hasil penelitian ini diharapkan dapat menjadi bahan referensi untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Analisis fitur audio dibagi menjadi 2 (dua) yaitu fitur audio berdasarkan domain frekuensi dan fitur audio berdasarkan domain waktu, fitur audio berdasarkan domain frekuensi meliputi MFCC, *Spectrogram*, *Chromagram*, *Spectral Centroid*, dan *Spectral Roll Off*, sedangkan fitur audio berdasarkan domain waktu meliputi *Zero Crossing Rate* dan *Energy* (Jondya & Iswanto, 2018). Perbedaan mendasar fitur audio domain frekuensi dan domain waktu yaitu domain frekuensi menyajikan plot berupa frekuensi terhadap magnitudo, sedangkan domain waktu menyajikan plot berupa variable waktu terhadap amplitudo. Amplitudo pada domain waktu menunjukkan keras lemahnya sinyal yang diterima, sedangkan frekuensi menunjukkan tingkat perubahan frekuensi sinyal.

Proses ekstraksi fitur bermula dari audio yang diubah menjadi array angka, kemudian hasil ekstraksi yang telah disesuaikan menjadi array 2 dimensi akan menjadi *input* metode klasifikasi. Metode klasifikasi yang digunakan dapat berupa *machine learning* atau *deep learning*. Beberapa metode yang pernah digunakan diantaranya CNN untuk klasifikasi 4 aksen yaitu *Arabic*, *Italian*, *Japanese* dan *Korean*, namun penelitian tersebut belum memasukkan aksen *English*, hasil percobaan dengan menggunakan fitur MFCC dan klasifikasi CNN menyimpulkan bahwa performa terbaik adalah aksen Italian dengan nilai akurasi mencapai 86,9 % (Chionh et al., 2018). Selain itu, fitur MFCC dan *Energy* juga dapat dipasangkan

dengan metode klasifikasi KNN untuk menganalisa pengucapan huruf vocal yang diproduksi oleh pembicara beraksen *American*, *Hindi* dan *Mandarin* (Patel & Barkana, 2018), percobaan tersebut menghasilkan bahwa MFCC dan *Energy* merupakan fitur terbaik dengan aksen Hindi yang paling mudah dikenali dengan akurasi sebesar 79,55 %. Penggunaan 5 fitur audio seperti MFCC, *Spectrogram, Chromagram, Spectral Centroid, dan Spectral Roll Off* juga pernah dilakukan percobaan dengan metode CNN untuk klasifikasi 5 aksen yaitu *Arabic*, *English*, *French*, *Mandarin*, dan *Spanyol*, hasil dari percobaan tersebut menyimpulkan bahwa MFCC-CNN mampu mencapai akurasi sebesar 48,24% untuk segment paragraph dan 53,92 % untuk segment 3 kata (Singh et al., 2019). Nilai akurasi yang rendah membuat peneliti yang sama melakukan percobaan lagi dengan membandingkan antara 5 kelas (*English, Spanish, Arabic, Mandarin* dan *French*) dan 3 kelas (*Arabic, English* dan *mandarin*), hasil perbandingan menyimpulkan bahwa akurasi yang dicapai dari 3 kelas lebih tinggi yaitu sebesar 71,43 % (Singh et al., 2020). Sifat aksen yang beragam dan karakteristik pengucapan yang tidak menentu (aksen tidak memiliki pola pasti) membuat akurasi yang dicapai masih cukup rendah, sehingga penelitian tentang aksen masih cukup menantang untuk dapat dilakukan lagi demi tercapainya performa yang lebih baik.

Klasifikasi audio menggunakan CNN umumnya menerima *input* berupa gambar *spectrogram*, sebuah penelitian membandingkan penggunaan ekstraksi fitur (fitur-SVM) dan *input* gambar *spectrogram* (image-CNN), hasilnya penggunaan ekstraksi fitur memberikan akurasi yang lebih tinggi, namun masih

menggunakan metode *machine learning* yaitu SVM (Theodore Giannakopoulos et al., 2019).

Tabel 2. 1 menjabarkan matriks *literature review* dan posisi penelitian, berdasarkan penelitian yang telah ada, metode ekstraksi fitur yang paling banyak digunakan adalah MFCC, karena mirip dengan system persepsi pendengaran manusia (dapat menangkap karakteristik fonetik, misalnya pengucapan e dan *e*), selain itu juga berdasarkan beberapa penelitian, MFCC mampu mencapai nilai akurasi yang lebih baik dibandingkan ekstraksi fitur lainnya (Chionh et al., 2018; Singh et al., 2019, 2020), tetapi metode ekstraksi fitur berdasarkan domain waktu seperti ZCR dan *Energy* belum pernah dilakukan percobaan pada penelitian aksen. Metode ZCR dan *Energy* pernah dilakukan percobaan analisa huruf vokal dan mampu memberikan performa yang baik (Barkana & Patel, 2020; Patel & Barkana, 2018). CNN merupakan salah satu metode *deep learning* yang dapat digunakan untuk *speech recognition*, walaupun secara umum CNN banyak digunakan untuk klasifikasi *image recognition* yang memerlukan *input* 3 dimensi, tetapi CNN juga dapat digunakan untuk klasifikasi audio yang umumnya berukuran 1 dimensi. Agar CNN dapat digunakan dalam klasifikasi audio, diperlukan beberapa tahapan untuk mempersiapkan audio menjadi *input* yang dapat diterima oleh CNN (Theodore Giannakopoulos et al., 2019; Hernandez et al., 2018). Oleh karena itu, penelitian ini bertujuan menganalisa ekstraksi fitur untuk menjadi *input* CNN dan merekomendasikan ekstraksi fitur mana (ZCR atau *Energy*) yang dapat memberikan performa terbaik terhadap klasifikasi CNN.

2.2. Keaslian Penelitian

Tabel 2. 1 Matriks *Literatur Review* dan Posisi Penelitian
Analisis Ekstraksi Fitur Audio pada *Convolutional Neural Network* untuk Pengenalan Aksen

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Application of *Convolutional Neural Networks* in Accent Identification	Chionh et al., Carnegie Mellon University www.andrew.cmu.edu, 2018	Mengklasifikasi 4 aksen yaitu Arabic, Italian, Japanese dan Korean.	Fitur MFCC dan CNN dapat digunakan untuk klasifikasi 4 aksen, dari ke empat aksen, aksen Italian mampu mencapai nilai akurasi tertinggi yaitu 86,9 %.	Belum memasukkan aksen English.	Penelitian ini masih menggunakan 4 aksen dan belum memasukkan aksen English, sehingga penulis akan memasukkan aksen English.
2	Features of speech audio for *deep learning* accent recognition	Singh et al., CEUR Workshop Proceeding http://ceur-ws.org/ , 2019	Mengklasifikasi 5 aksen yaitu English, Spanish, Mandarin, French, Arabic.	Membandingkan 5 fitur audio yaitu MFCC, Spectrogram, Chromagram, Spectral Centroid, dan Spectral Roll Off menggunakan metode CNN. MFCC-CNN memberikan kurasi terbaik yaitu 48,24% (segment paragraph) dan 53,92 % (segment 3 kata).	Belum menggunakan fitur ZCR dan *Energy*	Penelitian ini belum menggunakan fitur ZCR dan *Energy*, sehingga penulis menggunakan fitur ZCR dan *Energy* dan metode CNN.

Tabel 2.1 Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Features of Speech Audio for Accent Recognition	Singh et al., International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, 2020	Membandingkan antara 5 kelas (English, Spanish, Arabic, Mandarin dan French) dan 3 kelas (Arabic, English dan mandarin.	Hasil perbandingan menyimpulkan bahwa akurasi yang dicapai dari 3 kelas lebih tinggi yaitu sebesar 71,43 %.	Penelitian ini mengungkapkan bahwa penelitian tentang *aksen recognition* merupakan sebuah tantangan, karena semakin banyak kelas maka semakin kecil akurasi. Hal ini dikarenakan belum ada pola pasti dalam pengucapan (voice print).	Penelitian sebelumnya menggunakan 3 kelas dan 5 kelas, sedangkan penulis menggunakan 6 kelas menggunakan fitur yang berbeda yaitu *Energy* dan ZCR. Penelitian ini juga bertujuan meningkatkan akurasi dari penelitian sebelumnya walaupun dengan menambah kelas.
4	Analysis of American English Corner Vowels Produced by Mandarin, Hindi, and American Accented Speakers and Baseline Accent Recognition System	Patel & Barkana, Long Island Systems, Applications and Technology Conference, 2018.	Menganalisa huruf vocal yang dihasilkan oleh pembicara Mandarin, Hindi dan Amerika.	Fitur MFCC dan *Energy* memberikan akurasi yang baik yaitu 69,55 %	Metode yang digunakan KNN.	Penelitian ini masih menggunakan KNN, sedangkan penulis menggunakan CNN.

Tabel 2.1 Matriks *Literatur Review* dan Posisi Penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Analysis of vowel production in Mandarin/Hindi/American-accented English for accent recognition systems	Barkana & Patel, Applied Acoustics Elsevier, 2020	Menganalisa huruf vocal oleh pembicara Mandarin, Hindi dan Amerika. menggunakan fitur MFCC, *Energy*, Zero-Crossing Rate dan klasifikasi KNN	Fitur MFCC dan *Energy* memberikan akurasi terbaik yaitu 75,76 %. Sedangkan ZCR 73,33 %.	Penggunaan fitur *Energy* dan ZCR sebatas analisa pengucapan huruf vocal, belum menganalisa aksen, dan metode yang digunakan KNN.	Penelitian ini menganalisa dataset pengucapan huruf vokal dan menggunakan metode KNN, sedangkan penulis menggunakan dataset aksen dan metode CNN.
6	*Deep learning* for Classficiation of Speech Accents in Video Games	Hernandez et al., CEUR Workshop proceeding, 2018.	Klasifikasi aksen British (71,2 %) dan American (70,9 %) dengan panjang segment 1 detik.	Image Spectrogram digunakan sebagai *input* yang akan diklasifikasikan.	Menggunakan image spectrogram sebagai *input*, belum menggunakan ekstraksi fitur.	Penelitian ini menggunakan image spectrogram sebagai *input*. Sedangkan penulis menggunakan ekstraksi fitur sebagai *input*.
7	Recognition of Urban Sound Events Using Deep Context-Aware Feature Extractors and Handcrafted Features	Theodore Giannakopoulos et al., Advances in Information and Communication Technology, 2019	Membandingkan *input* CNN (ekstraksi fitur dan spektrogram). hasil perbandingan adalah ekstraksi fitur memberikan hasil yang lebih baik.	Metode Ekstraksi Fitur lebih baik digunakan (69,4 %) dibandingkan *input* gambar spectrogram (65,1 %).	Pasangan metode yang digunakan yaitu ekstraksi fitur - SVM, dan image spectrogram –CNN.	Penelitian ini masih menggunakan SVM (*input* fitur+SVM), sedangkan penulis menggunakan CNN (*input* fitur + CNN)

2.3. Landasan Teori

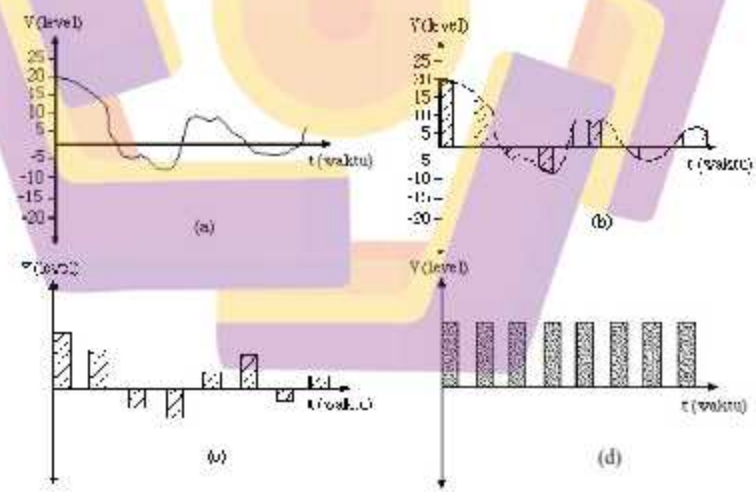
2.3.1. Berkas WAV

Terdapat beberapa format file audio yang paling umum digunakan, diantaranya adalah format Mp3 dan WAV. Masing-masing memiliki peran dalam dunia audio digital. Format audio mempengaruhi kualitas pemutaran, format dengan kualitas tinggi akan terdengar bagus tetapi menghabiskan banyak ruang penyimpanan, sedangkan format dengan kualitas rendah dapat menghemat ruang penyimpanan walaupun memiliki bit rate yang rendah.

Dataset yang digunakan pada penelitian ini berasal dari *The Speech Accent Archive George Mason University* yang diunduh secara gratis di *http://accent.gmu.edu/* berisi berkas audio aksen dari berbagai negara dengan format audio MP3. Dataset tersebut menggunakan format MP3, seperti halnya yang banyak dilakukan oleh sebagian besar perangkat lunak lainnya lebih menyukai format MP3 karena mampu mendapatkan ukuran berkas yang rendah dan tidak memakan ruang penyimpanan. Dikarenakan format awal audio berupa MP3 maka penulis mengkonversi format MP3 ke WAV, hal ini dilakukan karena WAV merupakan format standar audio (Mcfee et al., 2015), selain itu format WAV memiliki keunggulan *lossless audio* yaitu menawarkan kualitas yang tinggi (*bit rate* tinggi) walaupun melalui proses kompresi (Hidayat et al., 2018), sehingga proses konversi MP3 ke WAV tidak akan mengubah kualitas audio.

2.3.2. Proses sampling

Proses sampling adalah proses pencuplikan dari bentukan sinyal analog. Terdapat suatu aturan dari proses sampling sinyal, berdasarkan Teori Shannon bahwa frekuensi sinyal paling sedikit adalah 2 kali frekuensi sinyal yang akan disampling, sinyal yang disampling ini adalah sinyal analog (Viranda, 2017), frekuensi tersebut merupakan batas minimum dari frekuensi sample agar nantinya cuplikan yang diambil menunjukkan bentukan sinyal yang asli, lebih besar frekuensi akan lebih baik, karena cuplikan akan lebih menggambarkan sinyal yang asli. Gambar 2. 1 menunjukkan proses pencuplikan sinyal, setelah dilakukan proses sampling maka terbentuklah suatu sinyal analog-diskrit yang bentuknya menyerupai aslinya namun hanya diambil diskrit-diskrit (mencirikan) saja.



Gambar 2. 1 Proses Sampling Sinyal Analog Menjadi Sinyal Diskrit

Gambar 2. 1 (a) merupakan bentukan gelombang suara dari sebuah audio atau biasa disebut sinyal analog, sedangkan Gambar 2. 1 (b) dan (c) merupakan proses sampling sinyal analog, proses sampling diperlukan untuk mengubah sinyal analog menjadi bit-bit digital agar audio dengan mudah dapat diproses kembali. Gambar 2. 1 (d) merupakan proses quantisasi yaitu pembandingan level-level tiap diskrit sinyal hasil sampling dengan tetapan level tertentu, sinyal-sinyal diskrit akan disesuaikan levelnya, jika lebih kecil akan dinaikkan dan jika lebih besar akan diturunkan menyesuaikan tetapan yang ada, proses ini hampir sama dengan pembulatan angka. Setelah diquantisasi maka tiap-tiap diskrit yang ada telah memiliki tetapan tertentu. Tetapan ini dapat dijadikan kombinasi bilangan biner, maka terbentuklah bilangan-bilangan biner yang merupakan informasi dari sinyal.

2.3.3. Ekstraksi fitur audio

Ekstraksi fitur adalah proses pengambilan ciri dari suatu sinyal audio yang dapat merepresentasikan informasi penting dari sinyal audio yang selanjutnya dapat dimanfaatkan untuk analisis audio ataupun klasifikasi berkas audio (Perdana et al., 2017). Fitur audio dibagi menjadi fitur audio berdasarkan domain waktu dan fitur audio berdasarkan domain frekuensi (Theodoros Giannakopoulos & Pikrakis, 2014). Fitur audio domain waktu diekstraksi langsung dari sample sinyal audio, sedangkan fitur audio domain frekuensi (spectral audio fitur) biasanya didasarkan pada *Discrete Fourir Transform* (DFT) sinyal audio. DFT digunakan untuk melakukan analisa frekuensi dari sinyal waktu, untuk menghitung fitur *spectral* harus dilakukan perhitungan DFT *frame* audio terlebih dahulu. *Discrete Fourier*

Transform (DFT) adalah prosedur yang digunakan dalam pemrosesan sinyal digital atau gambaran karakteristik spektrum dari suatu sample data. DFT memungkinkan untuk menganalisa, memanipulasi dan mensintesis sinyal yang tidak mungkin dapat dilakukan dalam pemrosesan sinyal analog. Untuk membedakan fitur audio domain waktu dan domain frekuensi dapat dilihat pada table 2.2 berikut:

Tabel 2. 2 Penjelasan Fitur Audio Domain Waktu dan Domain Frekuensi

No	Fitur Audio	Penjelasan	Kesimpulan
A	Fitur Audio Domain Waktu		
	1. Short Time *Energy*	Fitur yang menunjukan nilai untuk mengukur kecenderungan apakah suara termasuk kedalam kategori voiced speech atau unvoiced speech. Voiced yaitu penyuaraan berupa vokal dan unvoiced berupa penyuaraan berupa konsonan.	Short time *Energy* umumnya digunakan untuk membedakan voiced atau unvoiced speech.
	2. *Zero Crossing Rate* (ZCR)	Zero crossing merupakan fitur yang nilainya menunjukkan jumlah berapa kali nilai amplitudo pada bit data sinyal melewati nilai 0. Suara yang tergolong unvoiced speech memiliki nilai ZCR yang lebih besar, sedangkan suara voiced speech memiliki nilai ZCR yang lebih kecil (Bachu et al., 2010)	ZCR umumnya digunakan untuk membedakan voiced atau unvoiced speech.
B	Fitur Audio Domain Frekuensi		
	1. Spectral Centroid	Spectral Centroid merupakan fitur yang merepresentasikan titik pada spectrum, dimana sebagian besar energi terpusat di titik spectrum tersebut. Spectral centroid sering dikaitkan dengan tingkat kejelasan spektral, dimana semakin tinggi nilai spectral centroid maka suara yang ada akan semakin tajam dan jelas.	Spectral centroid umumnya digunakan pada analisis sinyal music.
	2. Spectral Flux	Spectral Flux merupakan fitur yang mengukur seberapa cepat energi dari sinyal berubah. Perhitungan spectral flux dilakukan dengan membandingkan energi sinyal pada sebuah frame dengan frame sebelumnya sehingga dapat mengukur	Spectral Flux pernah dilakukan penelitian klasifikasi genre music.

Tabel 2. 2 Penjelasan Fitur Audio Domain Waktu dan Domain Frekuensi (Lanjutan)

No	Fitur Audio	Penjelasan	Kesimpulan
		perbedaan spektral dari frame ke frame dan dapat mengkarakteristikan perubahan bentuk spectrum.	
	3. Spectral Rolloff	Sama halnya dengan spectral centroid, spectral rolloff digunakan untuk mengukur ketajaman spectral. Hasil perhitungan spectral rolloff dapat digunakan untuk mengukur apakah suara termasuk dalam voiced speech ataukah unvoiced speech. Voiced speech pada umumnya memiliki nilai spectral rolloff yang lebih kecil apabila dibandingkan dengan unvoiced speech.	Spectral Roll Off pernah dilakukan penelitian klasifikasi genre music.
	4. Spectral Flatness	Spectral flatness merupakan fitur yang menunjukan karakteristik dari sebuah spektrum audio	Spectral flatness pernah dilakukan penelitian klasifikasi genre music.
	5. MFCCs	MFCC merupakan tipe representasi cepstral dari sinyal, dimana pita frekuensi didistribusikan menurut skala-mel, bukan pada pendekatan jarak linier. Umumnya, koefisian MFCC yang dihitung adalah 13.	Metode ini paling banyak digunakan karena Mirip dengan system persepsi pendengaran manusia yaitu dapat menangkap karakteristik fonetik (misalnya pengucapan e dan *e*)
	6. Chroma Vector	Chroma Vector merupakan representasi 12 unsur dari spectral *Energy*.	Chroma Vector banyak digunakan pada aplikasi yang berkaitan dengan musik.

Tabel 2. 2 menjelaskan definisi masing-masing ekstraksi fitur dan penggunaannya di berbagai penelitian. Fitur audio domain frekuensi (*Spectral Centroid, Spectral Flux, Spectral Rolloff, Spectral Flatness, MFCCs*, dan *Chroma Vector*) banyak digunakan untuk klasifikasi berkas musik, karena untuk membedakan berkas musik diperlukan analisis sinyal audio berdasarkan frekuensi. Sedangkan, fitur audio domain waktu (*short time Energy* dan ZCR) banyak digunakan untuk klasifikasi suara berdasarkan *voiced* (huruf vocal) atau *unvoiced speech* (konsonan). Hal ini dikarenakan *short time Energy* dan ZCR dapat mengekstraksi fitur berdasarkan besar-kecilnya amplitudo. Dataset aksen

merupakan sekumpulan audio suara yang mengucapkan huruf vokal dan konsonan, sehingga pemilihan fitur *Short time Energy* dan ZCR terhadap dataset aksen merupakan pemilihan fitur yang tepat dan dapat merepresentasikan audio aksen.

2.3.4. *Energy*

Energy adalah salah satu ekstraksi fitur sinyal audio berdasarkan domain waktu yang dapat dikatakan seberapa tinggi/ intensitas sinyal audio (Bachu et al., 2010). *Energy* dapat dihitung dengan persamaan (Theodoros Giannakopoulos & Pikrakis, 2014) (1)

$$E(i) = \frac{1}{W_L}\sum_{n=1}^{WL}|x_i(n)|^2 \quad (1)$$

Keterangan :
E : nilai fitur short time *Energy*
WL : jumlah sample yang ada pada frame i
X(n) : data sinyal pada window dengan panjang WL

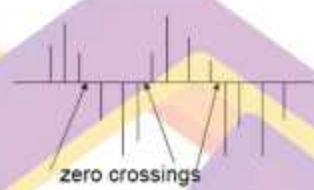
Root Mean Square Energy dari sinyal dapat dihitung dengan persamaan berikut (Knees & Schedl, 2016) (2) :

$$E(i) = \sqrt{\frac{1}{W_L}\sum_{n=1}^{WL}|x_i(n)|^2} \quad (2)$$

2.3.5. *Zero Crossing Rate*

Zero Crossing Rate termasuk ke dalam metode ekstraksi fitur sinyal audio berdasarkan domain waktu. *Zero Crossing Rate* dapat dikatakan jumlah berapa kali

dalam interval waktu/ bingkai sebuah amplitudo melawati nilai nol (Theodoros Giannakopoulos & Pikrakis, 2014). *Zero Crossing Rate* dapat diartikan sebagai ukuran kebisingan dari suatu sinyal, biasanya ZCR menunjukkan nilai yang lebih tinggi dalam kasus sinyal noice. Definisi *Zero Crossing Rate* dapat diilustrasikan pada Gambar 2. 2.



Gambar 2. 2. Definisi *Zero Crossing Rate*

Persamaan *Zero Crossing Rate* dapat dilihat pada persamaan (3) (4) berikut:

$$Z_i = \frac{1}{2W_L}\sum_{n=1}^{W_L} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (3)$$

Dimana sgn adalah

$$sgn[xi(n)] = \begin{cases} 1, xi(n) \geq 0, \\ -1, xi(n) < 0 \end{cases} \quad (4)$$

Keterangan :

Z : nilai *Zero Crossing*
X(n) : nilai amplitude pada data ke-n
WL : jumlah total bit yang ada pada frame
sgn : apabila lebih dari sama dengan 0 maka 1, dan apabila kurang dari 0 maka -1

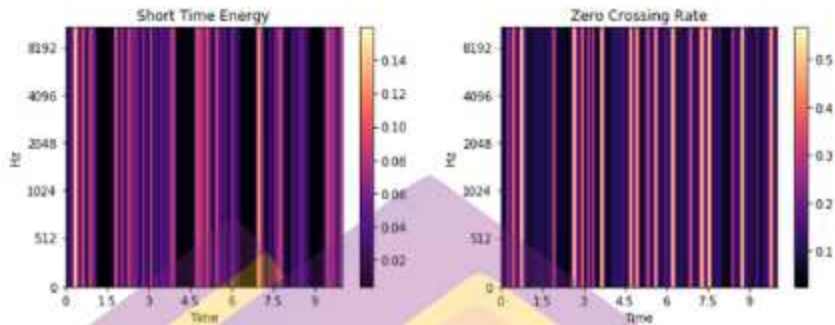
2.3.6. *Spectrogram*

Suara dihasilkan melalui dua buah proses yaitu *Generation* dan *Filtering*. Proses *generation* yaitu pemrosesan awal dari suara yang akan diproduksi melalui

bergetarnya pita suara (*vocal cord* dan *vocal fold*) yang berada di laring untuk menghasilkan bunyi periodik. Bunyi periodik bersifat konstan dan disaring melalui *vocal tract* (juga disebut dengan istilah resonator suara atau *articulator*) yang terdiri dari lidah, gigi, bibir, dan langit-langit mulut sehingga bunyi tersebut dapat menjadi bunyi keluaran (*output*) berupa bunyi vokal (*vowel*) dan atau bunyi konsonan (*consonant*) yang membentuk kata-kata yang memiliki arti yang nantinya dapat dianalisa untuk *voice recognition*.

Spectrogram adalah representasi *spectral* (warna suara) yang bervariasi terhadap waktu yang menunjukkan tingkat *density* (intensitas energi) spektral. Dengan kata lain *spectogram* adalah bentuk visualisasi dari masing-masing nilai formant yang dilengkapi dengan level energi yang bervariasi terhadap waktu. Level energi dikenal dengan istilah formant bandwidth, Formant adalah frekuensi-frekuensi resonansi dari filter, yaitu *vocal tract* (*articulator*) yang meneruskan dan memfilter bunyi keluaran (*output*) berupa kata-kata yang memiliki makna. *Spectogram* memuat hal-hal yang bersifat detil, oleh beberapa ahli *spectrogram* dikenal dengan istilah sidik suara (*voice print*) (Al-Azhar & Nuh, 2011).

Librosa adalah salah satu *library* yang dapat menampilkan *spectrogram* suara, tampilan *spectrogram* yang dihasilkan menunjukkan hubungan antara frekuensi dan waktu, serta tingkat intensitas suara yang ditunjukkan dengan warna spectrogram, semakin terang maka semakin tinggi intensitas suara, sedangkan semakin gelap maka semakin rendah tingkat intensitas suara. Gambar 2. 3 merupakan hasil *display spectrogram short time Energy* (kiri) dan *Zero Crossing Rate* (kanan) dengan durasi 10 detik.



Gambar 2. 3. *Display Spectrogram Short Time Energy* (Kiri) dan *Zero Crossing Rate* (Kanan)

Contoh hasil ekstraksi fitur dari salah satu file audio ditunjukkan pada Tabel 2. 3 dan Tabel 2. 4 berikut. Hasil esktraksi fitur yang ditampilkan pada table 2.3 dan 2.4 adalah fitur dari 1 detik pertama dari keseluruhan durasi, fitur berukuran 2 dimensi dengan bentuk 1,44. Bentuk 1,44 didapatkan dari *sample_rate*/ *hop_length* (*sample_rate* dibagi *hop_length*). Pengaturan nilai *sample_rate* dan *hop_length* mengikuti *default* librosa yaitu 22050/512 = 43,06 yang dibulatkan menjadi 44 fitur.

Tabel 2. 3 Hasil Ekstraksi Fitur *Short Time Energy*

[[0.02301641 0.02760891 0.02851335 0.02886692 0.03226678 0.02873597
0.02797642 0.03258276 0.03224776 0.03878803 0.04826006 0.05503755
0.06206337 0.0901411 0.13786776 0.15468785 0.15973575 0.14824156
0.11846434 0.09548858 0.08036805 0.07186478 0.03844508 0.02762139
0.03033392 0.055698 0.07795314 0.09782243 0.10492611 0.09468602
0.07683081 0.04754522 0.03136494 0.03100479 0.02839843 0.02320684
0.0268922 0.06987579 0.08485894 0.09480626 0.1017561 0.08528101
0.07762539 0.07526446]]

Tabel 2. 4 Hasil Ekstraksi Fitur *Zero Crossing Rate*

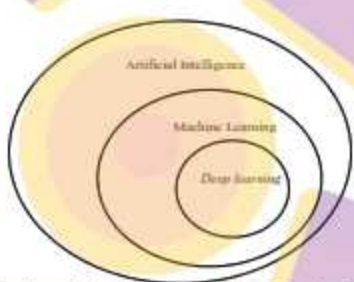
[[0.05419922 0.08300781 0.10839844 0.125 0.10449219 0.10058594
0.09619141 0.07324219 0.06201172 0.07714844 0.15527344 0.23486328
0.30126953 0.29541016 0.20410156 0.12451172 0.06689453 0.15576172
0.30322266 0.3984375 0.44677734 0.38232422 0.27392578 0.21337891
0.203125 0.16503906 0.13623047 0.10888672 0.07421875 0.09521484
0.20703125 0.32666016 0.45800781 0.56689453 0.51318359 0.41748047
0.32666016 0.19091797 0.14306641 0.13183594 0.10644531 0.11328125
0.0859375 0.05419922]]

2.3.7. *Convolutional Neural Network*

Deep learning (Pembelajaran mendalam) adalah salah satu metode pembelajaran mesin (*machine learning*) yang memungkinkan komputer untuk mempelajari tugas-tugas yang disesuaikan dengan sifat manusia (Nilam et al., 2019). Belakangan ini, *Deep learning* berhasil mengungguli performansi *machine learning* setelah pada Tahun 2012 *Deep learning* tampil pada kompetisi pengenalan citra *ImageNet Large Scale Visual Recognation Competition* (ILSVRC) menjadi pemenang dengan akurasi tertinggi. Sejak saat itu, *deep learning* mendapatkan perhatian khusus pada dunia komputasi.

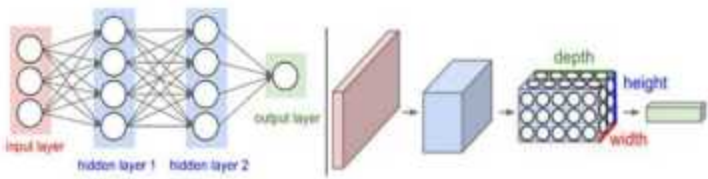
Deep learning memiliki hubungan yang erat dengan *machine learning* dan *artificial intelligence*, karena *deep learning* merupakan bagian dari *machine* learning dan *machine learning* merupakan bagian dari *artificial intelligence* (Ahmad, 2017), seperti diilustrasikan pada gambar 3.

Gambar 2. 4 menunjukkan bahwa *Artificial Intelligence*, *Machine Learning*, dan *Deep learning* merupakan irisan yang saling berhubungan, *Artificial Intelligence* merupakan kemampuan mesin untuk meniru perilaku kecerdasan manusia. Sedangkan, *Machine Learning* merupakan aplikasi dari kecerdasan buatan yang memungkinkan sebuah sistem untuk belajar secara mandiri (otomatis) dan meningkatkan kecerdasannya berdasarkan pengalaman pembelajaran, dan *Deep learning* merupakan aplikasi dari mesin pembelajaran dengan menggunakan metode yang lebih kompleks dan jaringan saraf mendalam pada tahapan pembelajaran suatu model.



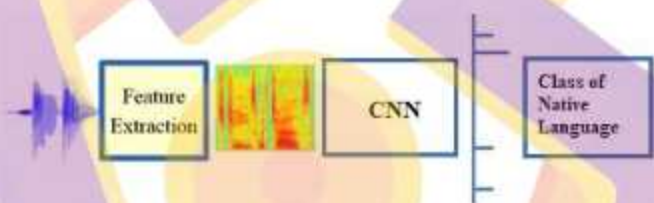
Gambar 2. 4. Hubungan AI, ML, dan DL

Salah satu Metode *Deep learning* yang banyak digunakan adalah *Convolutional Neural Network* (CNN). CNN mengatur neuron sehingga memiliki tiga dimensi (lebar, tinggi dan kedalaman) yang pada Gambar 2. 5 didefinisikan sebagai satu layer. Setiap layer pada CNN mentransformasi *input* 3D menjadi *output* 3D dari aktivasi neuron.



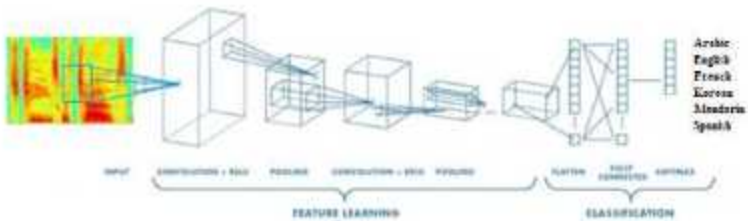
Gambar 2. 5. Jaringan Saraf Tiruan (kiri) dan CNN (kanan)

CNN tidak hanya digunakan pada pengenalan citra, tetapi juga dapat digunakan untuk pengenalan suara (*Speech Recognition*). Cara kerja *Speech Recognition System* dijelaskan pada Gambar 2. 6 berikut.



Gambar 2. 6. *Speech Recognition System*

Gambar 2. 6 menjelaskan bahwa sebuah sinyal suara akan diekstraksi fitur dengan bantuan metode tertentu, metode ekstraksi fitur akan mengubah sinyal suara menjadi data atau angka-angka, selain itu, untuk memudahkan analisis suara, metode ekstraksi fitur juga mengubah sinyal suara menjadi sebuah spectrogram yang memvisualisasikan sinyal suara tersebut, sehingga level *Energy* dari sinyal suara dapat dianalisis. Setelah melalui tahap esktraksi fitur, data tersebut akan dilakukan proses klasifikasi dengan bantuan metode CNN. Arsitektur CNN dijelaskan pada Gambar 2. 7 berikut :



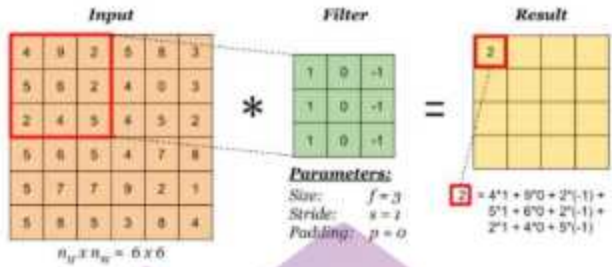
Gambar 2. 7. Arsitektur CNN

Sama halnya dengan object gambar yang menjadi *input* CNN, pada *speech recognition* (Gambar 2. 6), *input* CNN adalah spectrogram yang telah direpresentasikan dengan angka-angka. *Input* tersebut kemudian memasuki tahap *feature learning* dimana tahapan ini terdiri dari *convolutional layer* dan *pooling layer* (Hariyani et al., 2020).

Convolutional Layer

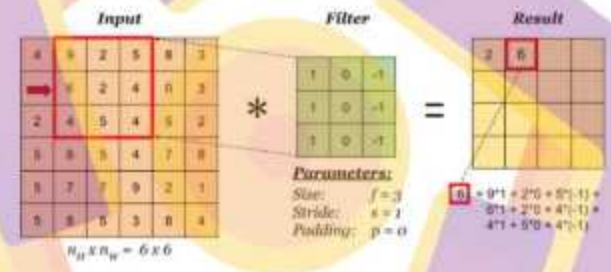
Convolutional layer yaitu proses konvolusi (*scanning*) antara *filter matrix* dengan *input matrix*. Langkah-langkah operasi konvolusi sebagai berikut (Nurhikmat, 2018) :

1. Lakukan perkalian antar *input* (kotak merah), dengan bobot *filter*, seperti yang dijelaskan pada Gambar 2. 8 berikut :



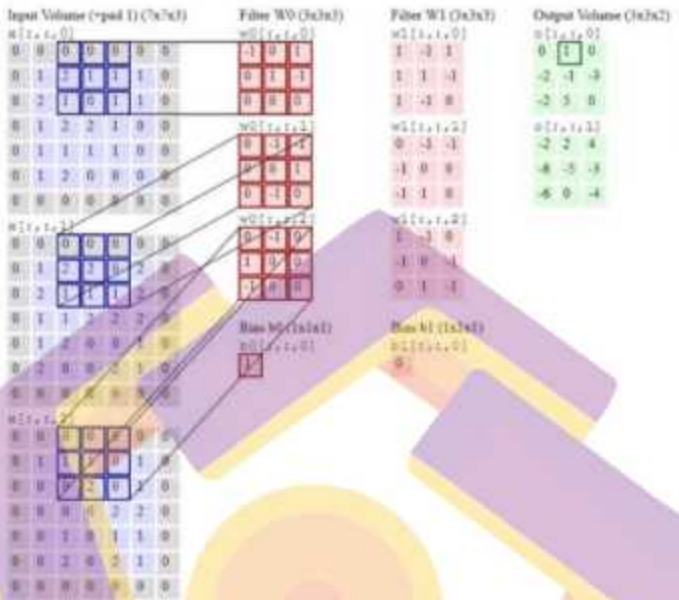
Gambar 2. 8. Proses konvolusi

2. Geser ke kanan 1 langkah berdasarkan pengaturan stride (contoh stride=1), dan lakukan lagi perkalian antar *input* (kotak merah), dengan bobot *filter*, seperti pada Gambar 2. 9 berikut :



Gambar 2. 9. Proses konvolusi berdasarkan *stride*

Stride adalah pengaturan berapa banyak pergeseran *cell* pada *input*. Pada Gambar 2. 10 pengaturan stride adalah 1 sehingga pergeseran *cell* kotak merah 1 *langkah*, lain halnya jika stride = 2, maka pergeseran cell kotak merah menjadi 2 langkah.



Gambar 2. 10. Proses konvolusi berdasarkan stride = 2

Proses konvolusi juga terdapat *padding*, yaitu penambahan *cell* bernilai 0 pada tepi gambar. *Padding* memiliki kelebihan dalam membangun jaringan yang lebih dalam, jika tidak dilakukan *padding* ada kemungkinan tinggi/lebar akan menyusut saat memasuki jaringan yang lebih dalam. *Padding* juga memiliki kelebihan menyimpan lebih banyak informasi di tepi gambar, tanpa padding dikhawatirkan sangat sedikit lapisan yang dipengaruhi oleh tepi gambar. Tujuan dari penggunaan padding adalah:

1. Dimensi *output convolutional* layer selalu lebih kecil dari *input*nya, *output* ini akan digunakan kembali sebagai *input* dari *convolutional* layer selanjutnya,

sehingga makin banyak informasi yang terbuang. Padding dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan convolutional layer yang lebih dalam/deep sehingga lebih banyak features yang berhasil diekstrak.

2. Meningkatkan performa dari model karena *convolutional* filter akan fokus pada informasi yang sebenarnya yaitu yang berada diantara zero padding tersebut.

Contoh penggunaan padding, jika terdapat dimensi *input* sebenarnya adalah 5x5, dan dilakukan convolution dengan filter 3x3 serta stride sebesar 2, maka akan didapatkan *feature map* dengan ukuran 2x2. Namun jika kita tambahkan zero padding sebanyak 1, maka *feature map* yang dihasilkan berukuran 3x3 (lebih banyak informasi yang dihasilkan).

Untuk menghitung dimensi feature map bisa digunakan persamaan 5 :

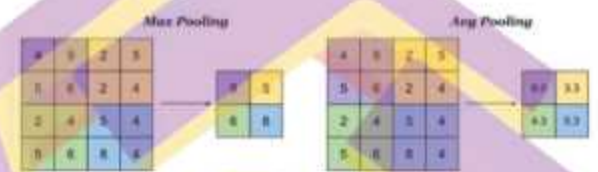
$$output = \frac{W - N + 2P}{S} + 1 \quad (5)$$

Keterangan :
W = Panjang / Tinggi *Input*
N = Panjang / Tinggi Filter
P = Zero Padding
S = Stride

Pooling layer terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan bergeser pada seluruh area feature map. Pooling layer digunakan untuk mengurangi ukuran spasial dengan tujuan mengurangi jumlah parameter dan komputasi, Jenis pendekatan *pooling* yang banyak digunakan yaitu *max pooling*

dan *average pooling*. *Max pooling* mengambil nilai maksimum pada daerah tertentu, sedangkan *average pooling* mengambil nilai rata-rata.

Misalnya jika kita menggunakan *Max Pooling* 2x2 dengan stride 2, maka pada setiap pergeseran filter, nilai maximum pada area 2x2 pixel tersebut yang akan dipilih, sedangkan Average Pooling akan memilih nilai rata-ratanya. Seperti terlihat pada Gambar 2. 11.



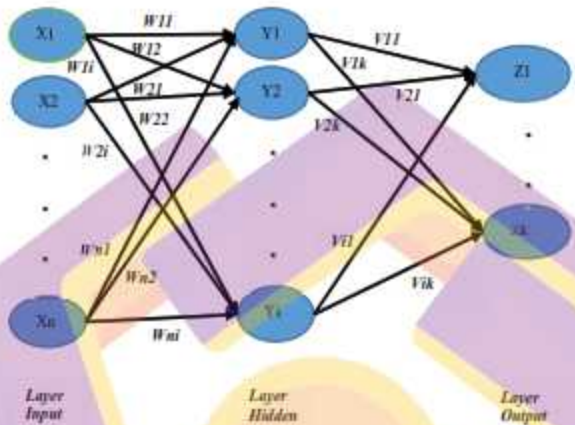
Gambar 2. 11. Max Pooling (kiri) dan Average Pooling (kanan)

Fully Connected Layer

Fully connected layer adalah jaringan syaraf tiruan *feedforward* yang terdiri dari *input layer*, *hidden layer* dan *output layer*, dimana setiap neuron pada suatu layer terhubung secara penuh (Haykin, 2009). *Feature map* yang akan dihasilkan oleh *feature extraction layer convolutional* masih berbentuk *multidimensional* array, sehingga harus dilakukan *"flatten"* atau *reshape feature map* menjadi sebuah vektor agar bisa digunakan sebagai *input fully connected layer*. *Fully Connected Layer* dijelaskan pada gambar Gambar 2. 12.

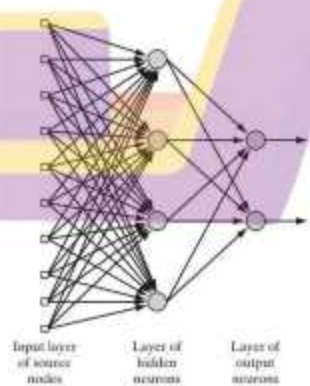
Arsitektur pada Gambar 2. 12 dikatakan *fully connected layer* karena antara layer *input* dan layer *hidden* terhubung penuh. Layer *hidden* terdiri dari neuron *hidden* yang melakukan perhitungan dari layer *input* ke layer *output*, layer *hidden* boleh lebih dari satu tergantung dengan kasus atau masalah yang akan diselesaikan. Layer *input* dihubungkan ke layer *hidden* menggunakan bobot, begitu juga dengan layer *hidden* ke layer *output*. Gambar 2. 12 memiliki 3 layer yaitu layer *input*, layer *hidden* dan layer *output*. Layer *input* memiliki *Xn* neuron, sedangkan layer *hidden*

memiliki *Yi* neuron, serta layer *output* memiliki *Zk* neuron. Antara layer *input* dan layer *hidden* dihubungkan dengan bobot (*Wni*), sedangkan antara layer *hidden* dengan layer *output* dihubungkan dengan bobot (*Vik*).



Gambar 2. 12 *Fully Connected Layer*

Sebagai contoh, jaringan pada Gambar 2. 13 memiliki arsitektur 10-4-2 karena jaringan tersebut memiliki 10 neuron pada *input* layer, 4 neuron pada *hidden* layer, dan 2 neuron pada *output* layer, masing-masing layer saling terhubung.



Gambar 2. 13 Jaringan *Feedforward* dengan arsitektur 10-4-2

2.3.8. *Confusion Matrix*

Untuk menentukan performa yang baik dari suatu model klasifikasi *machine learning* dapat menggunakan *confusion matrix* (Xu et al., 2020). *Confusion matrix* merupakan tabel nilai kombinasi dari nilai prediksi dan nilai actual untuk menghitung *accuracy, recall, precision,* dan *f1-score.*

Predicted Number

Actual Number						
		Class 1	Class 2	.	.	Class n
	Class 1	X_{11}	X_{12}	.	.	X_{1n}
	Class 2	X_{21}	X_{22}	.	.	X_{2n}
	.	.	.	.	.	.
	.	.	.	.	.	.
	Class n	X_{21}	X_{n2}	.	.	X_{nn}

Gambar 2. 14 *Confusion Matrix for Multiple Classes*

Gambar 2. 14 merupakan *table confusion matrix* untuk *multiple classes* (lebih dari 2 kelas), terdapat beberapa istilah dalam *confusion matrix* yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* adalah prediksi positif dan benar, *True Negative* adalah prediksi *negative* dan benar, *False Positif* adalah memprediksi positif dan salah, *False Negative* adalah memprediksi negative dan salah. Terdapat 4 (empat) parameter dari hasil perhitungan *confusion matrix* yaitu *accuracy, precision, recall* dan *F1-Score.* Perhitungan TP, TN, FP, dan FN pada *confusion matrix multiple classes* adalah dengan menjumlahkan TP keseluruhan, TN, FP, dan FN dari masing-masing kelas seperti persamaan 5, 6, 7, 8 berikut (Manliguez, 2016):

$$TTP_{all} = \sum_{j=1}^{n} x_{jj} \tag{5}$$

$$TTN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} x_{jk} \tag{6}$$

$$TFP_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ji} \quad (7)$$

$$TFN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} \quad (8)$$

Overall Accuracy adalah nilai yang menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. *Overall Accuracy* dapat dihitung dengan persamaan 9:

$$Overall\ Accuracy = \frac{TTP_{all}}{Total\ Number\ of\ Testing\ Entries} \quad (9)$$

Precision adalah nilai yang menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Precision* dapat dihitung dengan persamaan 10:

$$Precision_i = \frac{TTP_{all}}{TTP_{all} + TFP_i} \quad (10)$$

Recall adalah nilai yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Recall dapat dihitung dengan persamaan 11:

$$Recall_i = \frac{TTP_{all}}{TTP_{all} + TFN_i} \quad (11)$$

F1-Score adalah nilai rata-rata precision dan recall yang dibobotkan. F1-Score dapat dihitung dengan persamaan 12:

$$F1 - Score_i = \frac{2 * Recall_i * Precision_i}{Recall_i + Precision_i} \quad (12)$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah Eksperimen yaitu melakukan investigasi hubungan sebab akibat, apakah dengan menggunakan Ekstraksi Fitur *Zero Crossing Rate* (ZCR) atau *Energy* yang dipasangkan dengan *Convolutional Neural Network* (CNN) dapat meningkatkan performa (akurasi) dari sistem pengenalan aksen, dan apakah penggunaan fitur ZCR atau *Energy* dapat memberikan performa yang lebih baik dibandingkan penelitian sebelumnya.

Sifat penelitian ini adalah Kausal untuk mengeksplorasi hubungan sebab-akibat dan mendapatkan *output* berupa informasi atau pengetahuan dari penggunakan fitur ZCR atau *Energy* terhadap *Convolutional Neural Network*.

Pendekatan penelitian ini adalah Kuantitatif, karena memperbaiki nilai akurasi dari penelitian sebelumnya supaya mendapatkan akurasi yang lebih baik, penelitian ini melakukan pengujian fitur pada *Convolutional Neural Network* untuk mendapatkan sebuah kesimpulan berupa fitur terbaik.

3.2. Metode Pengumpulan Data

Penelitian ini menggunakan dataset publik yaitu *The Speech Accent Archive* dari *George Mason University* (http://accent.gmu.edu/) yang dapat diunduh secara gratis. Setiap audio merupakan rekaman dari pembicara yang berasal dari negara berbeda dengan mengucapkan kalimat yang sama yaitu *"Please call Stella. Ask her*

to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station."

Beberapa penelitian yang telah memanfaatkan dataset *The Speech Accent Archive* untuk pengenalan aksen diantaranya adalah penelitian pengenalan aksen dengan menggunakan fitur MFCC dan beberapa metode *Machine Learning* seperti GMM, K-Means, SVM, Naïve Bayes, Softmax dan GDA (Bryant et al., 2014), sedangkan baru-baru ini dataset tersebut telah dimanfaatkan dengan menggunakan banyak fitur (MFCC, *Spectrogram, Chromagram, Spectral Centroid*, dan *Spectral Roll-Off*) dan metode *deep learning* (CNN) (Singh et al., 2019) (Singh et al., 2020). Penelitian lainnya mengenai pengenalan aksen telah banyak dieksplorasi namun menggunakan dataset yang berbeda.

Dataset *The Speech Accent Archive* berisi ribuan rekaman suara dari berbagai negara di dunia dengan jumlah audio yang selalu bertambah, sampai saat laporan ini dibuat terdapat 2937 berkas audio yang dapat diunduh secara gratis untuk kepentingan analisis aksen. Namun penelitian ini dibatasi dengan mengambil 6 aksen dengan sampel terbanyak yaitu *Arabic, English, French, Korean, Mandarin, dan Spanish*, jumlah masing-masing aksen terlihat pada table 3.1 berikut:

Tabel 3. 1 Pemilihan Dataset

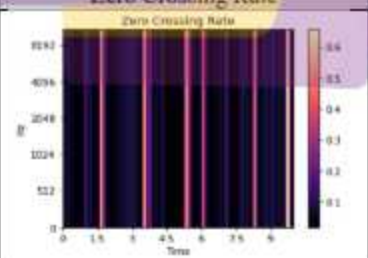
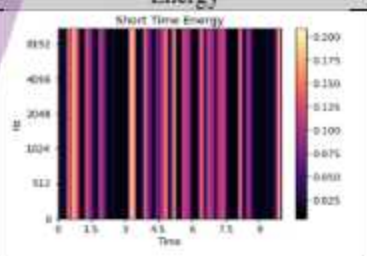
No	Native Language	Audio
1	Arabic	102
2	English	579

Tabel 3. 1 Pemilihan Dataset (Lanjutan)

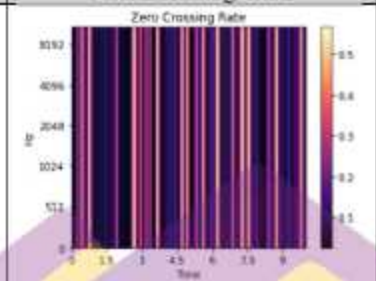
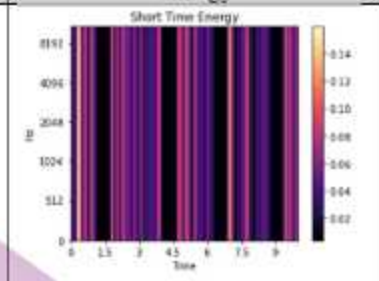
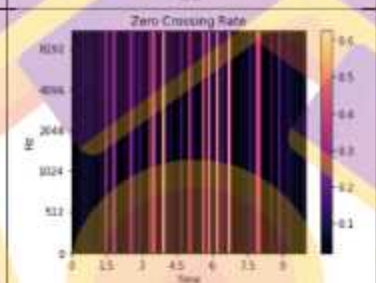
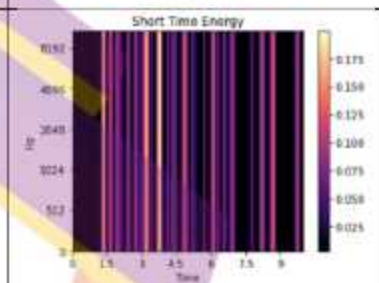
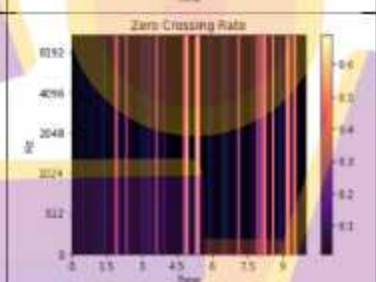
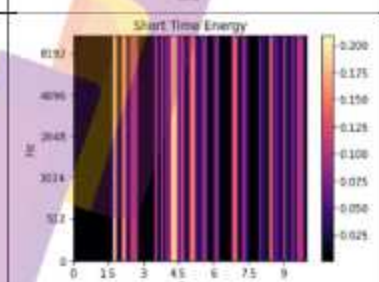
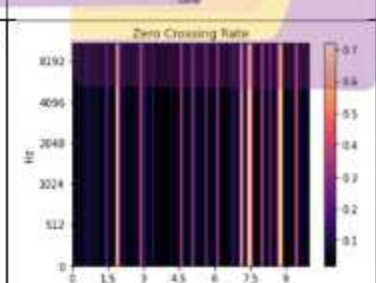
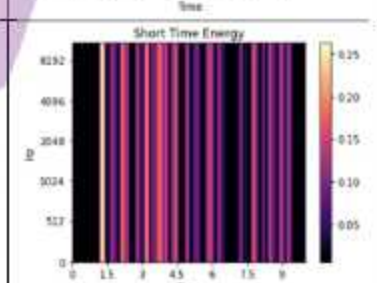
No	Native Language	Audio
3	French	63
4	Korean	52
5	Mandarin	65
6	Spanish	162
Total		1023

Karakteristik aksen dapat diketahui dengan melihat *display spectrogram (voice print)* dari setiap suara. *Spectrogram* menunjukkan tingkat intensitas suara yang ditandai dengan warna, semakin terang maka semakin tinggi intensitas suara, sedangkan semakin gelap maka semakin rendah tingkat intensitas suara. Tabel 3. 2 merupakan contoh *spectrogram* dari masing-masing kelas aksen, durasi (waktu) yang ditampilkan adalah 10 detik, dan frekuensi 8192 Hz, sedangkan intensitas suara ditandai dengan warna dan nilai yang ditunjukkan pada legenda di sebelah kanan, nilai intensitas suara ini lah yang nantinya akan menjadi nilai fitur (lihat Tabel 3. 3).

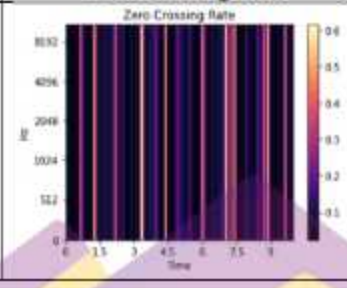
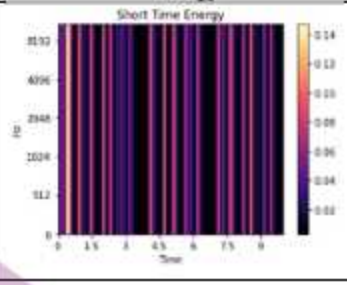
Tabel 3. 2 Contoh *Spectrogram* Masing-masing Aksen

File	Zero Crossing Rate	Energy
Arabic1.wav		

Tabel 3.2 Contoh *Spectrogram* Masing-masing Aksen (Lanjutan)

File	Zero Crossing Rate	Energy
English1.wav		
French1.wav		
Korean1.wav		
Mandarin1.wav		

Tabel 3.2 Contoh *Spectrogram* Masing-masing Aksen (Lanjutan)

File	Zero Crossing Rate	Energy
Spanish1.wav		

Tabel 3. 3 Contoh Nilai Fitur Masing-masing Aksen

File	Fitur ZCR	Fitur Energy
Arabic1.wav	[0.033203125, 0.0478515625, 0.0576171875, 0.05419921875, 0.05810546875, 0.0615234375, 0.07470703125, 0.06640625, 0.0693359375, 0.0673828125, 0.0556640625, 0.07080078125, 0.0693359375, 0.06103515625, 0.0634765625, 0.05126953125, 0.0498046875, 0.05810546875, 0.0517578125, 0.04921875, 0.03466796875, 0.02294921875, 0.0205078125, 0.01953125, 0.02636718750, 0.02636718750, 0.02636718750, 0.02685546875, 0.02099609375, 0.02197265625, 0.02294921875, 0.0234375, 0.02392578125, 0.02392578125, 0.02392578125, 0.0244140625, 0.0244140625, 0.02490234375, 0.02783203125, 0.03857421875, 0.06298828125, 0.0927734375, 0.0869140625, 0.06982421875]	[0.021382743492722 51, 0.021174175664782524, 0.024952743202447 89, 0.025416039419513 75, 0.024533607065677643, 0.024815396833419 8, 0.020250659435987473, 0.020020582625269 89, 0.020930379629135132, 0.021095002070069313, 0.021198099479079247, 0.019475510343909264, 0.020746493712067604, 0.022268507629632 95, 0.021160259031772614, 0.021712496876716614, 0.022731773555278778, 0.023694271221756935, 0.087503865536121368, 0.151396527886390 69, 0.163861633706092 83, 0.175370857119560 24, 0.158071979880332 95, 0.307001788914203 64, 0.099946841597557 07, 0.101676464080810 55, 0.123520873486995 7, 0.134767234325408 94, 0.165867105126380 92, 0.179366230964660 64, 0.187735229730606 08, 0.196163028478622 44, 0.179832741618156 43, 0.167620018124580 38, 0.154647350311279 3, 0.147648811340332 03, 0.149993598461151 2, 0.152334883809089 66, 0.141696587204933 17, 0.120266519486904 14, 0.091133601963520 05, 0.051183879375457764, 0.026056682690978 05, 0.024721313267946243]

Tabel 3.3 Contoh Nilai Fitur Masing-masing Aksen (Lanjutan)

File	Fitur ZCR	Fitur Energy
English1.wav	[0.05419921875, 0.0830078125, 0.1083984375, 0.125, 0.1044921875, 0.1005859375, 0.09619140625, 0.0732421875, 0.06201171875, 0.0771484375, 0.1552734375, 0.23486328125, 0.30126953125, 0.29541015625, 0.2041015625, 0.12451171875, 0.06689453125, 0.15576171875, 0.30322265625, 0.3984375, 0.44677734375, 0.38232421875, 0.27392578125, 0.21337890625, 0.203125, 0.1650390625, 0.13623046875, 0.10888671875, 0.07421875, 0.09521484375, 0.20703125, 0.32666015625, 0.4580078125, 0.56689453125, 0.51318359375, 0.41748046875, 0.32666015625, 0.1909179687 5, 0.1430664062 5, 0.1318359375, 0.1064453125, 0.11328125, 0.0859375, 0.05419921875]	[0.023016409948468 21, 0.027608910575509 07, 0.028513347730040 55, 0.028866918757557 87, 0.032266780734062 195, 0.028735965490341 187, 0.027976419776678 085, 0.032582756131887 436, 0.032247755676507 95, 0.038788031786680 22, 0.048260055482387 54, 0.055037554353475 57, 0.062063369899881 74, 0.090141102671623 23, 0.137867763638496 4, 0.154687851667404 17, 0.159733754132270 8, 0.143241564631462 1, 0.114646335799217 22, 0.095468578081130 98, 0.080368049442768 1, 0.071864783763885 5, 0.038443081561803 82, 0.027621394023299 217, 0.030333919450640 68, 0.055697996169328 69, 0.077953137457370 76, 0.097822427749633 79, 0.104926109139648 4, 0.094686023890972 14, 0.076830811798572 54, 0.047545224428176 88, 0.031364943832159 04, 0.031064786491394 043, 0.028398429974913 597, 0.023206839337944 984, 0.026892201974987 984, 0.069875791668899 19, 0.084858939051628 11, 0.094306261360645 3, 0.101756103336811 07, 0.085281014442443 85, 0.077625393867492 68, 0.075264461338520 05]
French1.wav	[0.02685546875, 0.0336914062 5, 0.0390625, 0.02880859375, 0.02490234375, 0.0234375, 0.02636718 75, 0.02978515625, 0.03466796875, 0.04345703125, 0.046875, 0.04589843 75, 0.0625, 0.05859375, 0.05322265625, 0.05517578125, 0.04589843 75,	[0.013412920758128166, 0.013403718359768 39, 0.011074499227106571, 0.007698222063481808, 0.006448643747717142, 0.005345334146473169, 0.005179757252335548, 0.005194804631173611, 0.005119195673614 74, 0.004491225816309452, 0.003960253205150366, 0.004500906914472 58, 0.005929501261562109, 0.007187218405306339, 0.008601783774793148, 0.008560319431126 18, 0.007659944240003824,

Tabel 3.3 Contoh Nilai Fitur Masing-masing Aksen (Lanjutan)

File	Fitur ZCR	Fitur Energy
French1.wav (Lanjutan)	0.04736328125, 0.04833984375, 0.0537109375, 0.064453125, 0.07568359375, 0.083984375, 0.07861328125, 0.0673828125, 0.0546875, 0.0537109375, 0.06494140625, 0.068359375, 0.0634765625, 0.0673828125, 0.06201171875, 0.0625, 0.07421875, 0.06591796875, 0.06103515625, 0.05126953125, 0.04638671875, 0.05810546875, 0.05908203125, 0.0556640625, 0.06396484375, 0.04345703125, 0.02929675]	0.0067258288487792, 0.0064592491835355576, 0.0062003857456147677, 0.0061745843850076 2, 0.006516828667372465, 0.006288821343332529, 0.007967489771544933, 0.008704511448740 96, 0.009597493335604668, 0.0093375938013196, 0.008268111385405064, 0.007690258789807558, 0.006516523659229278 6, 0.005918429233133793, 0.005332930944859981 5, 0.004893383942544 46, 0.004059033934026956 6, 0.003892050357535481 5, 0.003905335674062371 3, 0.003885275917127728 5, 0.003753788070753216 7, 0.003114367369562387 5, 0.003074052743613 72, 0.003086142474785447, 0.002789502264931798, 0.002701436402276158 3, 0.00260046171024441 7]
Korean1.wav	[0.04638671875, 0.0498046875, 0.0595703125, 0.052734375, 0.03466796875, 0.04638671875, 0.04541015625, 0.04052734375, 0.0390625, 0.033203125, 0.037109375, 0.0517578125, 0.04931640625, 0.052734375, 0.05419921875, 0.0556640625, 0.0546875, 0.0581054687 5, 0.06201171875, 0.0551757812 5, 0.068359375, 0.0629882812 5, 0.0581054687 5, 0.0625, 0.04833984375, 0.04833984375, 0.046875, 0.0458984375, 0.0532226562 5, 0.05419921875, 0.0532226562 5, 0.04736328125, 0.04833984375, 0.0561523437 5, 0.05419921875, 0.0615234375,	[0.0015675139147788286, 0.0014921160181984305, 0.0016003319760784507, 0.0017883174587 04114, 0.0017607519403100014, 0.0018866258906200528, 0.0018895812099799514, 0.0019019319443032146, 0.0019936750177294016, 0.0019653786439448595, 0.0020403780508 78644, 0.0018478196579962969, 0.0018678111955523 49, 0.0018644938245415688, 0.0017520511755719 78, 0.0018235576571896672, 0.0018009104533120 99, 0.0017023362452164292, 0.0016869688406586647, 0.0017572595970705152, 0.0018198932521045208, 0.0019359679426997 9, 0.0020257767755538225, 0.0020024173427373 17, 0.0020679328590631485, 0.0020442116074264 05, 0.0019411899847909808, 0.0024419482797384 26, 0.0025196399074047804, 0.0025473190471529 96, 0.0027665155939757824, 0.0025749602355062 96, 0.0024905505124479532, 0.0026996049564331 77, 0.0026874346658587456, 0.0025858965236693 62,

Tabel 3.3 Contoh Nilai Fitur Masing-masing Aksen (Lanjutan)

File	Fitur ZCR	Fitur Energy
Korean1.wav (Lanjutan)	0.06494140625, 0.05810546875, 0.05810546875, 0.05419921875, 0.06005859375, 0.06103515625, 0.05322265625, 0.03857421875]	0.0024840333499014378, 0.0022326286416500807, 0.0020646038465201855, 0.0018939013825729490, 0.0017780164489522576, 0.0017448231810703874, 0.0016613310435786843, 0.0015768057201057673]
Mandarin1.wav	[0.03271484375, 0.04638671875, 0.06103515625, 0.06884765625, 0.05322265625, 0.05078125, 0.04980468750, 0.04443359375, 0.05957031250, 0.05517578125, 0.06054687500, 0.06884765625, 0.06298828125, 0.07128906250, 0.05810546875, 0.04687500000, 0.04150390625, 0.03759765625, 0.04199218750, 0.04785156250, 0.05419921875, 0.05419921875, 0.06445312500, 0.07226562500, 0.07324218750, 0.08935546875, 0.09521484375, 0.09277343750, 0.09716796875, 0.08496093750, 0.07568359375, 0.07812500000, 0.07421875000, 0.07861328125, 0.07568359375, 0.07861328125, 0.07812500000, 0.07812500000, 0.08642578125, 0.07714843750, 0.06787109375, 0.06591796875, 0.04492187500, 0.02783203125]	[0.006495079491287470, 0.007181060966104269, 0.006751880981028080, 0.006633341312408447, 0.007386131677776575, 0.007319653406739235, 0.007363643962889910, 0.006904200185090303, 0.006197884221317935, 0.006602467969059944, 0.006498541217297316, 0.006638989318162203, 0.006508193910121918, 0.005673552397638559, 0.006407142151147127, 0.006680956110358238, 0.007127973716706037500, 0.006903569208085537, 0.007891944609582424, 0.007670242339372635, 0.008225922472774982, 0.007665840908885002, 0.007530836388468742, 0.007775290869176388, 0.007223047781735659, 0.007224825210869312, 0.007095555774867535, 0.007329397634837437, 0.007110778708010912, 0.007469027768820524, 0.007666039746254682500, 0.007044147234410047500, 0.006966044194996357, 0.006382093299180269, 0.006131853908300400, 0.006541691254824400, 0.006467231549322605, 0.006311672739684582, 0.005836638156324625, 0.005630644969642162, 0.006004104856401682, 0.005960426293313503, 0.006444115657359362, 0.006541965529322624]
Spanish1.wav	[0.03662109375, 0.04882812500, 0.05517578125, 0.05322265625, 0.05664062500, 0.05957031250, 0.06884765625, 0.07373046875, 0.06494140625,	[0.025619795545935630, 0.028821824118494987, 0.040593124926090240, 0.043398510664701460, 0.045094914734363556, 0.049562789499759674, 0.044764026999473570, 0.044508978724479675, 0.049122255295515060,

Tabel 3.3 Contoh Nilai Fitur Masing-masing Aksen (Lanjutan)

File	Fitur ZCR	Fitur Energy
Spanish1.wav (Lanjutan)	0.05712890625, 0.056640625, 0.04931640625, 0.04833984375, 0.06103515625, 0.05810546875, 0.05615234375, 0.0576171875, 0.0478515625, 0.052734375, 0.056640625, 0.05322265625, 0.052734375, 0.04541015625, 0.04052734375, 0.03662109375, 0.0498046875, 0.1015625, 0.21484375, 0.32519531125, 0.3896484375, 0.37255859375, 0.26904296875, 0.16845703125, 0.1162109375, 0.08447265625, 0.09033203125, 0.1005859375, 0.08740234375, 0.083984375, 0.06982421875, 0.05419921875, 0.04638671875, 0.038085937 5, 0.02734375]	0.046647068113088 61, 0.045447032898664474, 0.048371445387601 85, 0.043345026671886444, 0.040732629597187 04, 0.042651813477277756, 0.042292091995477676, 0.049231022596359 25, 0.085646040737628 94, 0.108513467013835 91, 0.133844718337059 02, 0.146783605217933 65, 0.134911924600601 2, 0.126524880528 45, 0.108741261065006 26, 0.093145087361335 75, 0.084773123264312 74, 0.067583514371090 09, 0.045707631856203 08, 0.027184393839740 753, 0.020613711327314 377, 0.016663074493408 203, 0.013969384333815 765, 0.011635145406830 31, 0.007689079269766 8076, 0.004244153853505 85, 0.004334442783147 097, 0.006503829732537 27, 0.009644380422139 168, 0.044589336961507 8, 0.071587406098542 62, 0.087643027305603 03, 0.096352614462375 64, 0.095103859901428 22, 0.090595476329326 63]

3.3. Metode Analisis Data

Setelah proses pengumpulan data, selanjutnya adalah proses analisis data untuk memastikan bahwa dataset yang dipilih sesuai dengan kebutuhan. Penelitian ini menggunakan seperangkat komputer dan *software* sebagai berikut:

1. Processor intel i5
2. RAM 8 GB
3. Harddisk 500 GB
4. Monitor, Mouse, dan Keyboard
5. Aplikasi Anaconda/ Spyder

6. *Library* librosa, Scikit Learn dan keras

Pada umumnya, CNN digunakan untuk klasifikasi citra yang membutuhkan *input* gambar berukuran 3 dimensi, sedangkan penelitian ini menggunakan *input* berupa audio yang harus dilakukan persiapan terlebih dahulu agar sinyal audio dapat menjadi *input* CNN 2D, untuk itu dibutuhkan sebuah *library* yang dapat mencirikan audio tersebut dan mengubah sinyal audio menjadi array berisi angka berukuran 2 dimensi atau lebih.

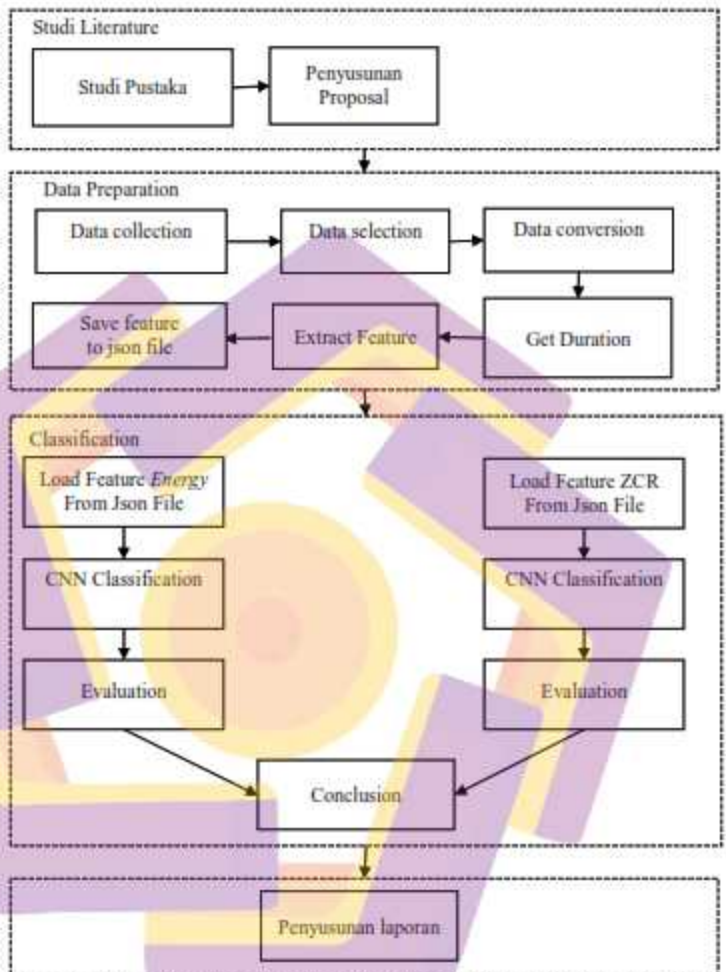
Salah satu *library* yang diperlukan untuk pemrosesan/ analisis audio menggunakan python adalah Librosa (Mcfee et al., 2015), librosa memungkinkan pengguna untuk membaca, mengekstrak, dan mendapatkan informasi yang terkandung di dalam berkas audio, sedangkan untuk pengembangan model CNN dan evaluasi model menggunakan *Scikit Learn* dan *Keras*.

3.4. Alur Penelitian

Alur penelitian yang akan dilakukan dapat dilihat pada Gambar 3. 1. Alur penelitian ini dimulai dari tahap studi pustaka dan penyusunan proposal. Setelah dilakukan studi pustaka mengenai artikel – artikel pengenalan aksen yang sudah ada, kemudian dilakukan penyusunan proposal agar diketahui tingkat *urgensi* penelitian yang akan dibuat. Setelah dilakukan penyusunan dan pengajuan proposal, tahap berikutnya yaitu menyiapkan dataset dan model klasifikasi.

Langkah – langkah menyiapkan dataset dan model klasifikasi sebagai berikut:

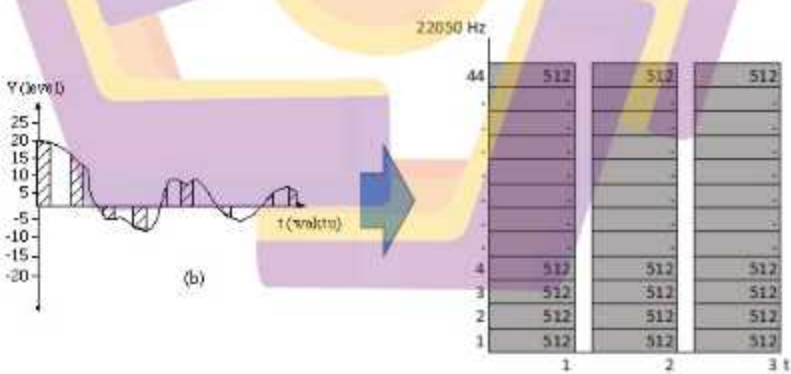
1. Pengumpulan dataset, diunduh dari *The Speech Accent Archive*;



Gambar 3. 1 Alur Penelitian

2. Pemilihan dataset berupa 6 aksen dengan jumlah sampel terbanyak (jumlah dataset dapat dilihat pada Tabel 3. 1);

3. Konversi data dari berkas asli yang berekstensi Mp3 ke WAV, tahap konversi ini dibutuhkan karena WAV merupakan format standar audio sehingga *library* librosa menggunakan parameter *input* berupa berkas WAV. Tahap konversi Mp3 ke WAV tidak mengubah kualitas audio karena perpindahan Mp3 ke WAV merupakan perpindahan dari bit rate rendah ke bit rate yang lebih tinggi.
4. *Get Duration* yaitu menentukan durasi audio yang akan diekstrak, pada penelitian ini akan dilakukan eksperiment dari panjang durasi 1 sampai 16 detik untuk mengetahui panjang durasi yang memberikan performa paling baik;
5. *Set Sampling Rate* yaitu menentukan nilai sample rate agar fitur yang diekstrak memiliki bentuk yang sama. Nilai *sample rate* yang digunakan pada penelitian ini mengikuti default dari librosa yaitu 22050 Hz, dan default *hop length* 512 Hz, nantinya bentuk yang didapatkan adalah nilai dari *sample_rate/ hop_length*. Proses penentuan betuk sinyal dijelaskan pada Gambar 3. 2 berikut :



Gambar 3. 2 Proses Penentuan Bentuk *Input* CNN

$$bentuk = \frac{22050}{512} = 44$$

Sehingga, *shape* yang didapatkan dari sampel sinyal adalah 44 setiap detiknya, Tabel 3. 4 menjelaskan penentuan bentuk pada setiap detiknya yang akan digunakan sebagai *input* CNN.

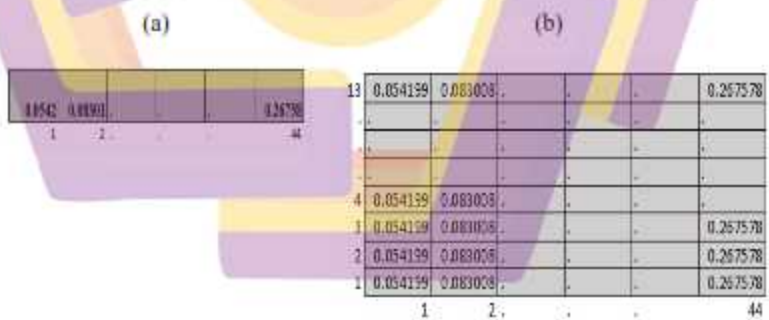
Tabel 3. 4 Penentuan Bentuk (*Shape*)

Durasi (detik)	Proses Penentuan Bentuk (*shape*)	Bentuk (*shape*)
1	$1\,x\,\frac{22050}{512}$	44
2	$2\,x\,\frac{22050}{512}$	87
3	$3\,x\,\frac{22050}{512}$	130
4	$4\,x\,\frac{22050}{512}$	173
5	$5\,x\,\frac{22050}{512}$	216
6	$6\,x\,\frac{22050}{512}$	259
7	$7\,x\,\frac{22050}{512}$	302
8	$8\,x\,\frac{22050}{512}$	345
9	$9\,x\,\frac{22050}{512}$	388
10	$10\,x\,\frac{22050}{512}$	431
11	$11\,x\,\frac{22050}{512}$	474
12	$12\,x\,\frac{22050}{512}$	517
13	$13\,x\,\frac{22050}{512}$	560
14	$14\,x\,\frac{22050}{512}$	603

Tabel 3. 4 Penentuan Bentuk (*Shape*) (Lanjutan)

Durasi (detik)	Proses Penentuan Bentuk (*shape*)	Bentuk (*shape*)
15	$15\ x\ \frac{22050}{512}$	645
16	$16\ x\ \frac{22050}{512}$	690

Tabel 3. 4 merupakan bentuk dari *input* yang akan disiapkan untuk klasifikasi CNN, *input* tersebut masih terlihat sebuah vektor 1 dimensi, dikarenakan CNN memerlukan *input* berupa array 2 dimensi atau lebih, maka diperlukan persiapan data agar *input* tersebut menjadi *input* array 2 dimensi. Pada penelitian ini, penulis akan menduplikat vektor tersebut menjadi array 2 dimensi dengan ukuran 13, 44 yaitu 13 baris dan 44 kolom, sedangkan dikarenakan sinyal audio merupakan *channel grayscale* maka ditambahkan *channel* 1 sehingga *input* shape menjadi 13, 44, 1 seperti Gambar 3. 3 berikut.

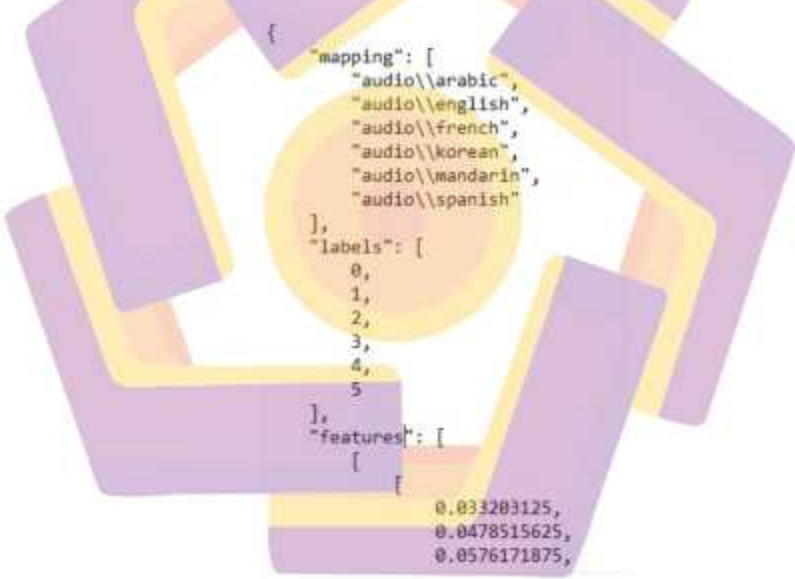


Gambar 3. 3 Persiapan *input* CNN berukuran 2 dimensi (Durasi 1 Detik)

Gambar 3. 3 (a) merupakan hasil ekstraksi fitur yang dikeluarkan oleh librosa dengan bentuk 1, 44 dan durasi 1 detik, sedangkan Gambar 3. 3 (b) merupakan

proses duplikasi vektor menjadi 13 baris sehingga menjadi bentuk 13, 44, agar *input* tersebut dapat diterima oleh CNN maka ditambahkan axis baru sehingga menjadi 13, 44, 1.

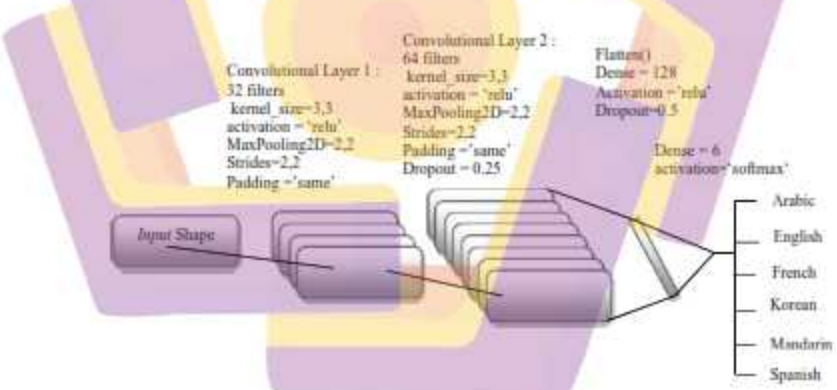
6. Setelah menentukan bentuk, proses selanjutnya yaitu proses ekstraksi atau proses pencirian sinyal audio, metode ekstraksi fitur yang digunakan yaitu *Energy* dan *Zero Crossing Rate* dengan library librosa, selanjutnya hasil ekstraksi fitur dari masing-masing metode disimpan di *file json* dengan struktur data *dictionary* seperti Gambar 3. 4.



Gambar 3. 4 *File json* dengan Struktur Data *Dictionary*

7. Klasifikasi yaitu langkah penentuan arsitektur CNN untuk mengklasifikasikan audio. Penentuan arsitektur CNN dilakukan berdasarkan beberapa kali percobaan seperti Tabel 3. 5, tidak ada aturan khusus mengenai parameter yang

digunakan untuk membuat arsitektur CNN termasuk berapa banyak *layer*, *filter*, atau ukuran parameter lainnya, yang diperlukan adalah pengalaman dalam membuat arsitektur sehingga menghasilkan arsitektur terbaik. Berdasarkan percobaan yang telah dilakukan, arsitektur terbaik dengan input 13, 44, 1, banyaknya *filter convolution layer* 1 adalah 32, ukuran filter 3,3, aktivasi relu, ukuran *maxPooling2D* 2,2, dan *filter convolutional layer 2* sebanyak 64, ukuran *filter* 3,3, aktivasi relu, *maxPooling2D* 2,2, *strides* 2,2, dan *dropout* 0.25, kemudian dilakukan *flatten* dengan *node* 128, aktivasi relu dan *dropout* 0.5, karena percobaan ini memiliki *output* 6 kelas, sehingga menggunakan 6 *nodes* dengan aktivasi *softmax* (Kim, 2017). Arsitektur terbaik berdasarkan hasil percobaan dijelaskan pada Gambar 3. 5.



Gambar 3. 5 Arsitektur CNN 2 Layer

Tabel 3. 5 Percobaan Parameter Arsitektur CNN

Percobaan	Layer 1							Layer 2							Flatten					Akurasi
	Filters	Kernel_size	Activation	Data_format	Input_shape	MaxPooling2D	Stride	Filters	Kernel_size	Activation	MaxPooling2D	Stride	Padding	Dropout	Dense	Activation	Dropout	Dense	Activation	
1	13	3,3	relu	channel_last	13,44,1	2,2	2,2	13	3,3	relu	2,2	2,2	same	-	13	relu	-	6	softmax	0.52
2	13	3,3	relu	channel_last	13,44,1	2,2	2,2	26	3,3	relu	2,2	2,2	same	-	52	relu	-	6	softmax	0.54
3	26	3,3	relu	channel_last	13,44,1	2,2	2,2	26	3,3	relu	2,2	2,2	same	-	26	relu	-	6	softmax	0.53
4	32	3,3	relu	channel_last	13,44,1	2,2	2,2	32	3,3	relu	2,2	2,2	same	-	32	relu	-	6	softmax	0.53
5	32	3,3	relu	channel_last	13,44,1	2,2	2,2	64	3,3	relu	2,2	2,2	same	-	64	relu	-	6	softmax	0.54
6	32	3,3	relu	channel_last	13,44,1	2,2	2,2	64	3,3	relu	2,2	2,2	same	-	128	relu	-	6	softmax	0.54
7	32	3,3	relu	channel_last	13,44,1	2,2	2,2	64	3,3	relu	2,2	2,2	same	0.25	128	relu	0.5	6	softmax	0.57
8	44	3,3	relu	channel_last	13,44,1	2,2	2,2	44	3,3	relu	2,2	2,2	same	-	44	relu	-	6	softmax	0.5
9	44	3,3	relu	channel_last	13,44,1	2,2	2,2	88	3,3	relu	2,2	2,2	same	-	176	relu	-	6	softmax	0.53
10	44	3,3	relu	channel_last	13,44,1	2,2	2,2	88	3,3	relu	2,2	2,2	same	0.25	176	relu	0.5	6	softmax	0.56
11	128	3,3	relu	channel_last	13,44,1	2,2	2,2	128	3,3	relu	2,2	2,2	same	-	128	relu	-	6	softmax	0.5
12	128	3,3	relu	channel_last	13,44,1	2,2	2,2	256	3,3	relu	2,2	2,2	same	-	256	relu	-	6	softmax	0.56
13	128	3,3	relu	channel_last	13,44,1	2,2	2,2	256	3,3	relu	2,2	2,2	same	-	512	relu	-	6	softmax	0.54
14	128	3,3	relu	channel_last	13,44,1	2,2	2,2	256	3,3	relu	2,2	2,2	same	0.25	512	relu	0.5	6	softmax	0.56
15	200	3,3	relu	channel_last	13,44,1	2,2	2,2	200	3,3	relu	2,2	2,2	same	-	200	relu	-	6	softmax	0.56
16	200	3,3	relu	channel_last	13,44,1	2,2	2,2	400	3,3	relu	2,2	2,2	same	-	400	relu	-	6	softmax	0.56

8. *Evaluation* yaitu mengukur performa dari model yang telah dibuat. Evaluasi model menggunakan *confusion Matrix* dan *library scikit learn*.
9. Kesimpulan yaitu tahapan membandingkan hasil evaluasi. Pada tahap ini akan ditarik kesimpulan yaitu pemilihan metode ekstraksi fitur terbaik diantara ZCR + CNN atau *Energy* + CNN.
10. Penulisan laporan hasil penelitian.

Metode CNN lebih dikenal dengan klasifikasi gambar sehingga di dalam CNN sendiri sudah terdapat *feature extraction* (lihat Gambar 3. 6), berbeda dengan klasifikasi audio, *input* yang berupa sinyal audio harus dilakukan persiapan terlebih dahulu agar sinyal audio tersebut dapat diterima CNN dan dilakukan klasifikasi.

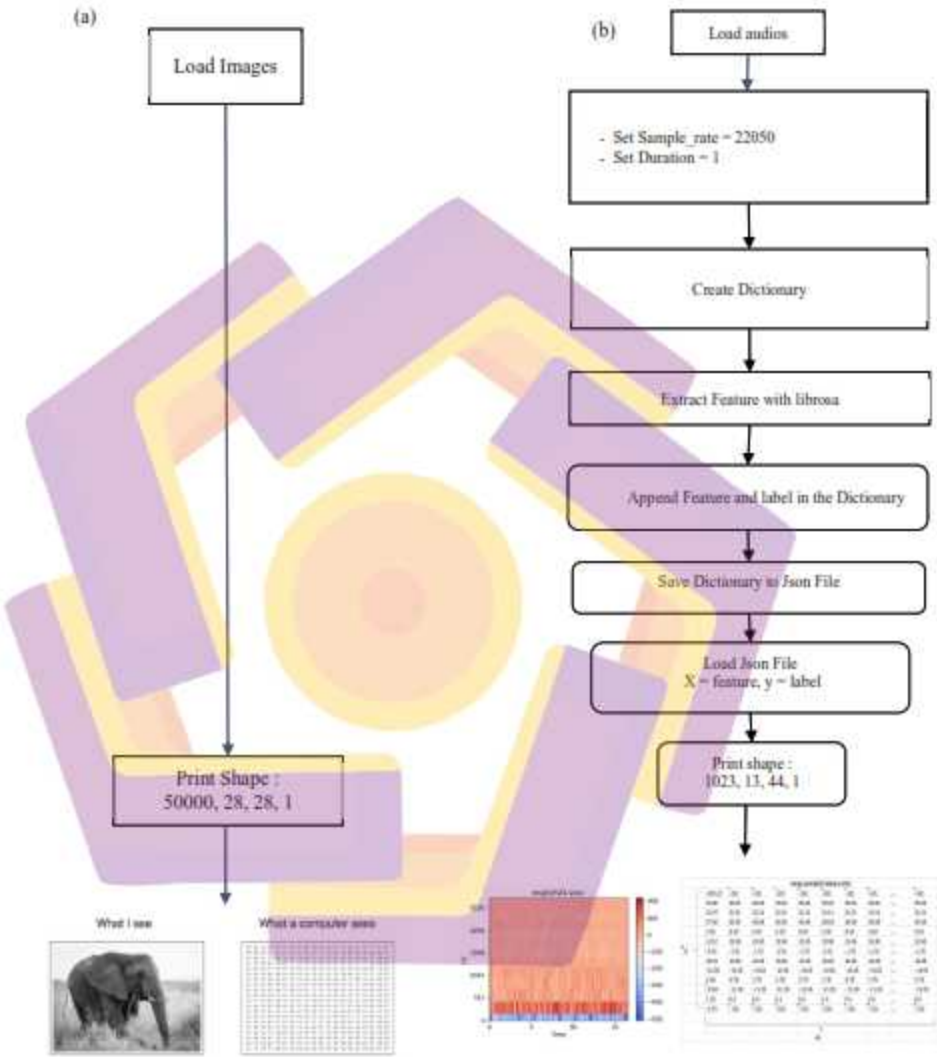
Gambar 3. 6 (a) menjelaskan bahwa ekstraksi fitur pada dataset gambar sudah dilakukan oleh CNN ketika dilakukan *load image*, system akan membaca pixel dari gambar, sehingga pada contoh terbentuk array yang berukuran 50000, 28, 28, 1 dimana 50000 yaitu banyaknya dataset, 28 baris pixel dan 28 kolom pixel, sedangkan angka 1 menunjukkan *channel grayscale*.

Gambar 3. 6 (b) adalah proses ekstraksi fitur audio yang melalui beberapa tahapan sehingga dihasilkan array yang dapat menjadi *input* CNN. Proses ekstraksi fitur audio dimulai dari *load* dataset audio, kemudian mengatur nilai *sample rate* sebesar 22050 Hz (*standard* librosa) agar setiap audio memiliki bentuk yang sama, dan menentukan durasi (contoh *set duration* 1 detik). Langkah selanjutnya yaitu membuat struktur data *dictionary* untuk menyimpan hasil ekstraksi fitur beserta labelnya, ketika proses ekstraksi fitur berjalan, hasil ekstraksi fitur disimpan

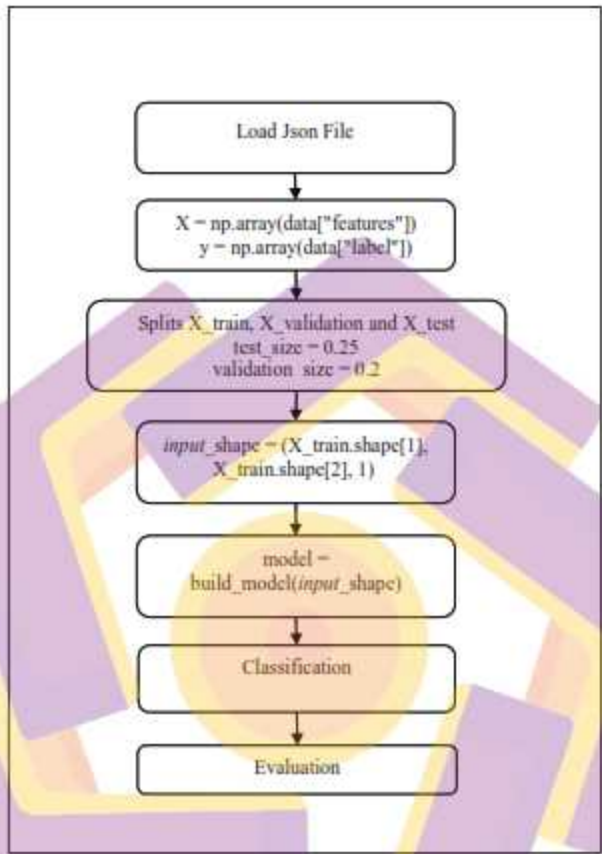
(*append*) ke dalam *dictionary*, dan disimpan dalam bentuk *file json*. *File json* yang berisi *feature* dan *label* telah siap menjadi input CNN.

Setelah memiliki *feature* dan *label* yang tersimpan di *file json*, selanjutnya adalah melakukan load *file json*, nilai feature yang ada pada *file json* akan disimpan pada *array X* dan nilai *label* akan disimpan pada *array y*, sehingga terdapat data *X* dan *y* yang telah siap menjadi *input* CNN dan siap dilakukan *split train-validation-test*, secara keseluruhan jika dilakukan *print_shape* maka bentuk *variable X* adalah 1023, 13, 44, 1 untuk 1 detik.

Tahap selanjutnya yaitu klasifikasi (lihat Gambar 3. 7), data *feature* akan disimpan pada *array X* dan data *label* akan disimpan pada *array y*, selanjutnya *split train-validation-test* dengan ukuran *test_size* = 0.25, *validation_size* = 0.2. *Input_shape* dari model yaitu mengambil index ke 1 dan ke 2 dari *X_train.shape* sedangkan nilai channel yaitu 1 karena hasil ekstraksi fitur merupakan array 2 dimensi sehingga disamakan dengan gambar *grayscale*, akhirnya *Input_shape* telah siap dan dapat menjadi *input* model CNN untuk dilakukan klasifikasi dan evaluasi.



Gambar 3. 6 (a) Ekstraksi Fitur pada Klasifikasi Gambar dan (b) Klasifikasi Audio



Gambar 3. 7 Tahap klasifikasi

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Persiapan Data

Persiapan data merupakan langkah menyiapkan fitur agar nantinya siap untuk dilakukan klasifikasi. Persiapan data dimulai dari pengumpulan data (*data collection*), pemilihan data (*data selection*), koversi data dari Mp3 ke Wav (*data conversion*), menentukan durasi (*get duration*), ekstraksi fitur audio (*extract feature*), dan menyimpan hasil ekstraksi fitur ke *file json* (*save feature to json file*).

4.1.1. Pengumpulan Data (*Data Collection*)

Dataset yang digunakan berasal dari *The Speech Accent Archive* dari *George Mason University* yang diunduh secara gratis di link *http://accent.gmu.edu/*.

4.1.2. Pemilihan Data (*Data Selection*)

Situs *The Speech Accent Archive* menyediakan ribuan audio yang merekam aksen dari berbagai negara seperti English, Jepang, Italia bahkan Indonesia. Penelitian ini mengambil 1023 audio dari 6 (enam) negara yang memiliki jumlah sample audio terbanyak, yaitu 102 *Arabic*, 579 *English*, 63 *French*, 52 *Korean*, 65 *Mandarin*, dan 162 *Spanish*,

4.1.3. Konversi Data (*Data Conversion*)

Format audio yang disediakan dari Situs *The Speech Accent Archive* berekstensi MP3, hal ini dapat dimaklumi karena file MP3 menawarkan *bit rate* yang rendah sehingga dapat menghemat banyak ruang penyimpanan. Librosa sebagai library yang membutuhkan format WAV karena WAV merupakan standar format audio dan menggunakan coding PCM (Pulse Code Modulation) sehingga semua sample audio disimpan semuanya di media penyimpanan dalam bentuk digital, memerlukan *input* yang berupa WAV agar dapat diproses, sehingga diperlukan konversi dari file asal (MP3) ke WAV. Proses konversi pada 1023 audio dilakukan satu per satu menggunakan situs konversi online di internet (*https://online-audio-converter.com/*). Proses konversi Mp3 ke Wav tidak akan mengubah kualitas dari audio dikarenakan tipe wav memberikan bit rate yang tinggi. Konversi Mp3 ke Wav dapat dilihat pada Gambar 4. 1 berikut :

Gambar 4. 1 Konversi audio dari Mp3 ke Wav

Gambar 4. 1 menjelaskan bahwa *file audio* dengan nama *arabic1* semula berekstensi Mp3 dan memiliki *bit rate* 711 KB (*bit rate* rendah), setelah dikonversi menjadi *file* berekstensi Wav, *bit rate file audio* arabic1 berubah menjadi 7.64 MB (*bit rate* tinggi), perubahan *bit rate* rendah menjadi tinggi justru lebih baik karena menawarkan ruang yang lebih besar.

4.1.4. Menentukan Durasi (*Get Duration*)

Masing-masing audio memiliki durasi yang berbeda-beda, rata-rata durasi berkisar Antara 16 – 54 detik, dengan durasi terpendek 16 detik, dan durasi terpanjang 54 detik, agar menjaga adanya panjang array yang sama antar tiap fitur audio, maka proses ekstraksi fitur audio dilakukan berdasarkan panjang durasi yang sama yaitu 1 sampai dengan 16 detik. Proses ekstraksi fitur dari durasi yang berbeda akan mengakibatkan panjang array yang berbeda (*shape* berbeda), perbedaan *shape* tidak dapat diterima oleh metode CNN.

4.1.5. Ekstrasi Fitur (*Extract Feature*)

Karena penelitian ini membandingkan fitur *Zero Crossing Rate* (ZCR) dan *Energy*, maka diperlukan 2 file json yang nantinya akan digunakan sebagai *input* CNN yaitu *data_zcr.json* dan *data_rms.json*. Fungsi yang digunakan untuk melakukan ekstraksi fitur menggunakan *library librosa*, seperti Gambar 4. 2 untuk fitur ZCR dan Gambar 4. 3 untuk fitur *Energy*. Fungsi ekstraksi fitur ZCR terdapat dalam fungsi *save_zcr* dan fungsi ekstraksi fitur *Energy* terdapat dalam fungsi *save_rms* yang akan dijelaskan pada sub bagian 4.1.6.

```
zcr = librosa.feature.zero_crossing_rate(signal[start:finish])
```

Gambar 4. 2 Ekstraksi Fitur *Zero Crossing Rate*

```
rms = librosa.feature.rms(signal[start:finish])
```

Gambar 4. 3 Ekstraksi Fitur *Root Mean Square Energy*

4.1.6. Penyimpanan Hasil Ekstraksi Fitur (*Save Feature to Json File*)

Hasil Ekstraksi fitur akan disimpan dalam *file json* dengan tipe data *dictionary*, untuk menyimpan hasil ekstraksi fitur ZCR akan dibuat fungsi yang bernama *save_zcr* dan untuk menyimpan hasil ekstraksi fitur *Energy* akan dibuat fungsi yang bernama *save_rms*. Kedua fungsi tersebut, memiliki *source code* yang sama, hanya berbeda pada bagian ekstraksi fitur seperti Gambar 4. 2 dan Gambar 4. 3.

Library yang dibutuhkan yaitu modul *json* untuk penyimpanan data, modul *os* untuk menampilkan letak direktori audio, modul librosa untuk mengekstrak audio, dan modul *numpy* untuk mengelola *vector* atau matriks. Selain itu, juga dilakukan inisiasi beberapa parameter yaitu *DATASET_PATH* sebagai letak direktori berkas audio, *JSON_PATH* sebagai nama *file json* yang akan digunakan untuk menyimpan fitur, menentukan *SAMPLE_RATE* dengan nilai 22050 (nilai ini merupakan standar dari librosa), menentukan durasi audio, dan nilai *SAMPLES* untuk proses *sampling* yang berguna dalam menentukan *shape* agar dapat digunakan sebagai *input* CNN. *Source Code* inisiasi parameter dijelaskan pada Gambar 4. 4 berikut.

```
DATASET_PATH = "audio"
JSON_PATH = "data_zcr.json"
SAMPLE_RATE = 22050
DURATION = 1 # duration in seconds
SAMPLES = SAMPLE_RATE * DURATION
```

Gambar 4. 4 Inisiasi Parameter

Setelah dilakukan inisiasi beberapa parameter, dibuatlah fungsi *save_zcr* untuk menyimpan hasil ekstraksi fitur dengan parameter *dataset_path* dan

json_path sebagai parameter *input* seperti terlihat pada Gambar 4. 5, di dalam fungsi *save_zcr* ini terdapat struktur data *dictionary* untuk menyimpan *mapping directory audio*, *labels audio*, dan *fitur zcr*.

```
def save_zcr(dataset_path, json_path):
    # dictionary to store mapping, labels, and features
    data = {
        "mapping": [],
        "labels": [],
        "zcr": []
    }
```

Gambar 4. 5 Fungsi Penyimpanan Ekstraksi Fitur

Setelah inisiasi *dictionary*, dilakukan *looping sub directory* yang ada pada *dataset_path*, kembaliannya yaitu *dirpath*, *dirnames* dan *filenames*, selanjutnya setiap nilai *dirpath* akan disimpan pada *dictionary* dengan kunci *mapping*. *Source Code looping sub directory* dijelaskan pada Gambar 4. 6.

```
# loop through all sub-dir
for i, (dirpath, dirnames, filenames) in enumerate(os.walk(dataset_path)):

    #if dirpath is not audio then enter sub dir with /
    if dirpath is not dataset_path:

        semantic_label = dirpath.split()[0]
        #output "mapping": ["audio\\arabic","audio\\english","audio\\french"
        data["mapping"].append(semantic_label)
```

Gambar 4. 6 *Looping* untuk *Mapping Sub Directory* Audio

Setelah melakukan *looping* pada setiap *sub directory* untuk menentukan/ *mapping* letak direktory audio, selanjutnya adalah *looping* semua berkas audio dan dilakukan ekstraksi fitur seperti Gambar 4. 7.

```
# looping all audio files in sub dir
for f in filenames:
    #example : audio\\arabic\\arabic1.wav and etc
    file_path = os.path.join(dirpath, f)
    signal, sample_rate = librosa.load(file_path, sr=SAMPLE_RATE)
    # calculate sample from start to finish
    start = 0
    finish = start + SAMPLES
    #extract feature, ex: 0 to 22050 for 1s, 0 to 66150 for 3s,
    #0 to 220500 for 10s
    zcr = librosa.feature.zero_crossing_rate(signal[start:finish])
    data["zcr"].append(zcr.tolist()"13)
    data["labels"].append(i-1)
```

Gambar 4. 7 Looping Semua Berkas Audio dan Ekstrak Fitur

Panjang audio yang diekstrak akan ditentukan berdasarkan nilai *SAMPLE* (22050 x durasi), contohnya *signal* ke 0 sampai 22050 (22050 x 1) untuk 1 detik, 0 sampai 44100 (22050 x 2) untuk 2 detik, 0 sampai 66150 (22050 x 3) untuk 3 detik, dan seterusnya, proses ini akan membentuk *shape* yang konsisten yaitu 44 untuk 1 detik, 87 untuk 2 detik, dan 130 untuk 3 detik, dan seterusnya. Konsep ini dapat dilihat pada Tabel 3. 4. Langkah terakhir yaitu menyimpan dictionary ke dalam *file json* seperti Gambar 4. 8.

```
# save data to json file
with open(json_path, "w") as fp:
    json.dump(data, fp, indent=4)
```

```
if __name__ == "__main__":
    save_zcr(DATASET_PATH, JSON_PATH)
```

Gambar 4. 8 Menyimpan fitur ke *file json*

Output dari fungsi *save_zcr* dan fungsi *save_rms* adalah *file json* seperti terlihat pada Gambar 4. 9.

(a)


```
{
    "mapping": [
        "audio\\arabic",
        "audio\\english",
        "audio\\french",
        "audio\\korean",
        "audio\\mandarin",
        "audio\\spanish"
    ],
    "labels": [
        0,
        1,
        2,
        3,
        4,
        5
    ],
    "zcr": [
        [
            [
                0.033203125,
                0.0478515625,
                0.0576171875,
```


(b)


```
{
    "mapping": [
        "audio\\arabic",
        "audio\\english",
        "audio\\french",
        "audio\\korean",
        "audio\\mandarin",
        "audio\\spanish"
    ],
    "labels": [
        0,
        1,
        2,
        3,
        4,
        5
    ],
    "rms": [
        [
            [
                0.0213827434927225,
                0.0211741756647825,
                0.0249527432024478
```


Gambar 4. 9 (a) *File json* Fitur *Zero Crossing Rate* dan (b) *File json* Fitur *Energy*

4.2. Klasifikasi

Setelah melakukan persiapan data yaitu melakukan ekstraksi fitur dari setiap 1, 2, 3 sampai dengan16 detik dimana menghasilkan 16 *file json* fitur ZCR dan 16 *file json* fitur *Energy*, langkah selanjutnya adalah membuat berkas klasifikasi menggunakan CNN. Klasifikasi diawali dengan memuat fitur (*load feature*) dari *file json* agar dapat dilakukan persiapan dataset.

4.2.1. Load Feature From Json File

Fungsi *load_data* adalah fungsi untuk memuat fitur dari *file json*, sebelumnya telah dijelaskan bahwa *file json* berisi *dictionary* dengan kunci fitur

(zcr atau *Energy*) dan *labels*, pada Gambar 4. 10 fungsi *load_data* akan melakukan *return array* X dari kunci *zcr* dan *array y* dari kunci *labels*, sehingga *array X* akan berisi fitur, dan *array y* akan berisi label dan siap digunakan sebagai dataset.

```
def load_data(data_path):

    #loads dataset from json file
    with open(data_path, "r") as fp:
        data = json.load(fp)
    #return X as fitur zcr
    #return y as labels
    X = np.array(data["zcr"])
    y = np.array(data["labels"])
    return X, y
```

Gambar 4. 10 Fungsi *Load Dataset*

4.2.2. Klasifikasi CNN

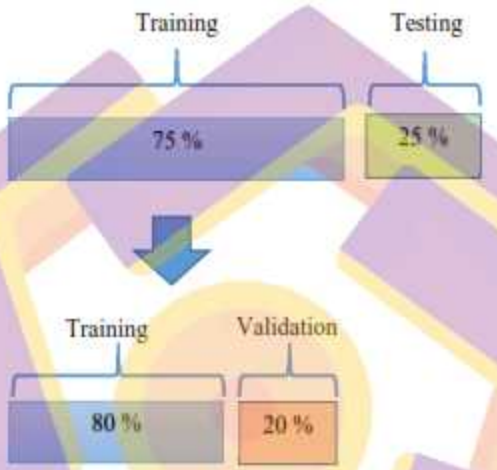
File klasifikasi diawali dengan inisiasi beberapa *library* diantaranya adalah *library json, numpy, train_test_split* dari *sklearn*, dan *keras* dari *tensorflow*, selain itu juga dilakukan inisiasi nama *file json* yang akan di-*import* yaitu *data_zcr.json*, seperti Gambar 4. 11.

```
import json
import numpy as np
from sklearn.model_selection import train_test_split
import tensorflow.keras as keras

DATA_PATH = "data_zcr.json"
```

Gambar 4. 11 Inisiasi Nama File Json yang akan di-*import*

Selanjutnya adalah melakukan persiapan dataset, yaitu membuat fungsi *prepare_dataset* untuk membagi/ *split* dataset menjadi *data testing*, *data training*, dan *data validation*. Parameter pada fungsi ini adalah *test_size* yaitu ukuran untuk

banyaknya *data testing* dan *validation_size* yaitu banyaknya *data validation*. Fungsi ini berguna untuk mengatur persentase pembagian dataset sesuai dengan skenario yang diinginkan. Skenario pembagian dataset dapat dilihat pada Gambar 4. 12 berikut :



Gambar 4. 12 Pembagian Dataset

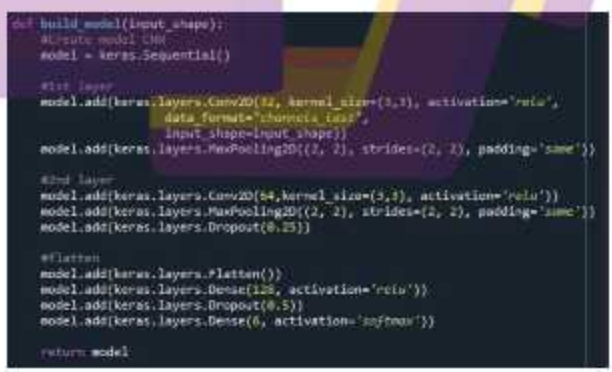
Gambar 4. 12 menjelaskan bahwa penelitian ini membagi dataset menjadi 25 % *data testing*, kemudian dari *data training*, diambil 20 % untuk *data validation*, sehingga jumlah *data testing* yaitu 256 data, *data validation* yaitu 154 data, dan *data training* 613 data dari total keseluruhan data sebanyak 1023 data.

Fungsi *prepare_datasets* juga menggunakan *Numpy newaxis* untuk menambah dimensi atau menambah angka 1 pada array, karena pemrosesan *signal* dianggap *grayscale*, sehingga diberi angka 1 di belakang array, berbeda dengan

pengenalan *image* karena RGB diberi angka 3. Fungsi *prepare_dataset* dapat dilihat pada Gambar 4. 13 berikut.

Gambar 4. 13 Fungsi *Prepare_dataset*

Fungsi berikutnya yaitu fungsi membuat model/ struktur CNN. Fungsi *build_model* memiliki parameter *input_shape* yaitu *shape* dari signal, misalnya untuk signal berdurasi 1 detik memiliki *shape* 13,44,1. Gambar 4. 14 memperlihatkan fungsi *build_model* untuk mengatur jumlah *layer, filter, kernel_size, activation, data_format, pooling, strides,* dan *padding*. Detail pemodelan CNN dapat pula dilihat pada Tabel 4. 1 berikut.



Gambar 4. 14 Membuat Model CNN

Tabel 4. 1 Pemodelan CNN

Layer 1	
Filters	32
Kernel_size	(3,3)
Activation	relu
Data_format	Channels_last
*Input*_shape	*Input*_shape
MaxPooling2D	(2,2)
Strides	(2,2)
Padding	same

Layer 2	
Filters	64
Kernel_size	(3,3)
Activation	relu
MaxPooling2D	(2,2)
Strides	(2,2)
Padding	same
Dropout	0.25
Flatten	
Dense	128
Activation	relu
Dropout	0.5
Dense	6
Activation	Softmax

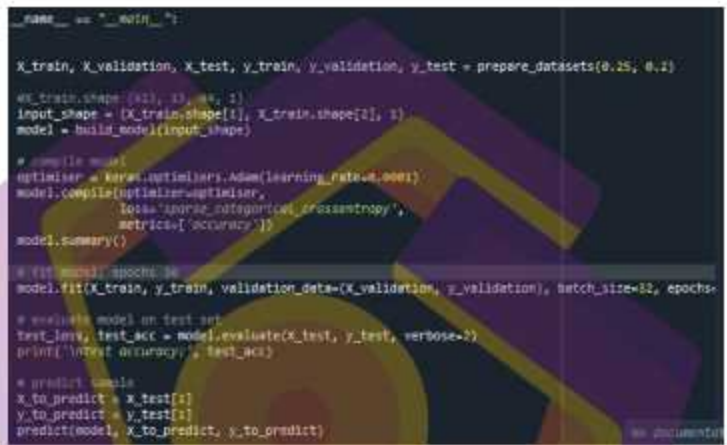
Fungsi berikutnya yaitu fungsi *predict* dengan parameter model, X dan y. yang berguna untuk memprediksi sebuah array X yang telah ditentukan. Fungsi *predict* dapat dilihat pada Gambar 4. 15 berikut.

```
def predict(model, X, y):
    # predict single sample
    # add a dimension to X, 15, 44, 1
    X = X[np.newaxis, ...]

    # perform prediction
    prediction = model.predict(X)
    #output[[0.06332318 0.54919904 0.0422731  0.02285489 0.02936492 0.29209567]]
    print("The prediction is", prediction)

    # get index with max value
    # for the example max value is 0.54919904 in index 1
    predicted_index = np.argmax(prediction, axis=1)

    print("Target: {}, Predicted label: {}".format(y, predicted_index))
    n = predicted_index
    y_prediksi.append(n)
    print('y_prediksi contains', y_prediksi)
```

Gambar 4. 15 Fungsi *Predict*

Cara kerja fungsi *predict* adalah dengan mencari nilai tertinggi dari hasil perhitungan model (menggunakan fungsi *np.argmax*). Sebagai contoh, variable *prediction* menghasilkan prediksi yaitu [[0.06332318 0.54919904 0.0422731

0.02285409 0.02939492 0.29295567]], dari array tersebut dipilih nilai yang tertinggi menggunakan fungsi *np.argmax* yaitu 0.54919904 yang berada di index 1, sehingga hasil prediksi berada di index 1, yang mana index 1 merupakan label dari kelas *English*.



Gambar 4. 16 Main Program

Gambar 4. 16 merupakan *main python* untuk menjalankan program dan memanggil fungsi – fungsi yang telah diinisiasi sebelumnya. Ukuran input shape yang digunakan diambil dari index ke 1, dan ke 2 dari x_train, sehingga menjadi 13,44,1 (contoh untuk durasi 1 detik). *Model.compile* digunakan untuk konfigurasi model, *model.fit* digunakan untuk melatih model, *model.predict* digunakan untuk memprediksi output dari sampel input, dan model.evaluate untuk menguji/ mengevaluasi model dengan mengembalikan nilai *loss* dan nilai *accuracy*.

Learning rate yang digunakan yaitu *0.0001* dan *epoch* 30. Untuk melakukan prediksi pada sebuah *sample data*, setiap fitur dan label pada *data testing (X_test*

dan y_test) dipindahkan sementara ke array *X_to_predict* dan *y_to_predict* barulah dilakukan prediksi dengan memanggil fungsi *predict*. Hal ini dilakukan untuk memudahkan dalam melakukan prediksi secara satu per satu, sehingga diketahui fitur, label sebenarnya, dan label hasil prediksi.

4.2.3. Ujicoba Hasil Prediksi

Ujicoba dilakukan dengan menjalankan fungsi *predict(model, X_to_predict, y_to_predict), sample data testing* dari fitur ZCR index ke- 0 dan index ke- 1 (shape 13, 44, 1) digunakan sebagai percobaan. Tabel 4. 2 menunjukkan hasil prediksi benar fitur index ke- 0 *data testing*, label sebenarnya yaitu 1 (kelas English), dan setelah diprediksi menghasilkan label 1 yang juga menunjukkan kelas English. Sedangkan Tabel 4. 3 menunjukkan hasil prediksi salah fitur index ke- 1 *data testing*, label sebenarnya yaitu 4 (kelas Mandarin), dan setelah diprediksi menghasilkan label 1 yang ternyata menunjukkan kelas English.

Tabel 4. 2 Contoh Hasil Prediksi Benar

Index ke-	Fitur	Label Sebenarnya	Label Hasil Prediksi	Keterangan
0	[[0.10058594] [0.14306641] [0.20214844] [0.19335938] [0.18554688] [0.18164062] [0.17333984] [0.1875] [0.19824219] [0.21582031] [0.20703125] [0.18652344] [0.18652344]	1	1	Label sebenarnya adalah 1 atau kelas English dan diprediksi **benar** dengan label 1 (kelas English)

Tabel 4. 2 Contoh Hasil Prediksi Benar (Lanjutan)

Index ke-	Fitur	Label Sebenarnya	Label Hasil Prediksi	Keterangan
	[0.17675781] [0.1796875] [0.19824219] [0.19824219] [0.18164062] [0.19628906] [0.17773438] [0.17578125] [0.18701172] [0.16992188] [0.17285156] [0.16992188] [0.15820312] [0.14453125] [0.12744141] [0.10058594] [0.07763672] [0.05615234] [0.04248047] [0.03417969] [0.04345703] [0.05419922] [0.05908203] [0.06591797] [0.06396484] [0.05957031] [0.05175781] [0.04003906] [0.03320312] [0.02099609] [0.01318359]]			

Tabel 4. 3 Contoh Hasil Prediksi Salah

Index ke-	Fitur	Label Sebenarnya	Label Hasil Prediksi	Keterangan
1	[[0.00121208] [0.00131514] [0.00134894] [0.00131665] [0.00147291] [0.00148369]	4	1	Label sebenarnya adalah 4 atau kelas Mandarin dan diprediksi **salah** karena menghasilkan label 1 (kelas English)

Tabel 4. 3 Contoh Hasil Prediksi Salah (Lanjutan)

Index ke-	Fitur	Label Sebenarnya	Label Hasil Prediksi	Keterangan
	[0.00141293] [0.00138416] [0.0012701] [0.001144] [0.00115113] [0.00119457] [0.00126001] [0.00130837] [0.00136431] [0.00141794] [0.00137599] [0.00144561] [0.0014578] [0.00142263] [0.00239547] [0.00328132] [0.00390719] [0.00461891] [0.01482957] [0.03180611] [0.04844882] [0.05366684] [0.05371488] [0.04800425] [0.03477806] [0.03281067] [0.03656767] [0.03572641] [0.03353969] [0.02822004] [0.01847062] [0.0137007] [0.00970358] [0.00735775] [0.008204] [0.04053396] [0.05885414] [0.05873502]]]			

4.3. Evaluasi

Salah satu pengukuran performa model adalah dengan mengetahui nilai loss dan akurasi yang dihasilkan, fungsi yang dijalankan yaitu *test_loss, test_acc = model.evaluate (X_test, y_test, verbose=2)* pada main phyton. Hasil evaluasi dari setiap fitur ZCR pada tiap durasi dijelaskan pada Tabel 4. 4, hasil evaluasi fitur ZCR mampu memperoleh akurasi tertinggi pada durasi 15 detik yaitu 62 % dan loss 1.32.

Tabel 4. 4 Ujicoba *Zero Crossing Rate* dan CNN

Durasi (detik)	Proses Penentuan Bentuk (*shape*)	Bentuk (*shape*)	Akurasi	Loss
1	$1\,x\,\frac{22050}{512}$	13, 44, 1	0.57	1.32
2	$2\,x\,\frac{22050}{512}$	13, 87, 1	0.59	1.25
3	$3\,x\,\frac{22050}{512}$	13, 130, 1	0.57	1.33
4	$4\,x\,\frac{22050}{512}$	13, 173, 1	0.59	1.25
5	$5\,x\,\frac{22050}{512}$	13, 216, 1	0.58	1.33
6	$6\,x\,\frac{22050}{512}$	13, 259, 1	0.59	1.28
7	$7\,x\,\frac{22050}{512}$	13, 302, 1	0.59	1.23
8	$8\,x\,\frac{22050}{512}$	13, 345, 1	0.58	1.35
9	$9\,x\,\frac{22050}{512}$	13, 388, 1	0.61	1.23
10	$10\,x\,\frac{22050}{512}$	13, 431, 1	0,59	1,3
11	$11\,x\,\frac{22050}{512}$	13, 474, 1	0.58	1.39

Tabel 4. 4 Ujicoba *Zero Crossing Rate* dan CNN (Lanjutan)

Durasi (detik)	Proses Penentuan Bentuk (*shape*)	Bentuk (*shape*)	Akurasi	Loss
12	$12\,x\,\frac{22050}{512}$	13, 517, 1	0.58	1.38
13	$13\,x\,\frac{22050}{512}$	13, 560, 1	0.58	1.33
14	$14\,x\,\frac{22050}{512}$	13, 603, 1	0.55	1.37
15	$15\,x\,\frac{22050}{512}$	13, 645, 1	0.62	1.32
16	$16\,x\,\frac{22050}{512}$	13, 690, 1	0.58	1.37

Gambar 4. 17 memperlihatkan grafik perolehan akurasi fitur ZCR dan CNN. Berdasarkan grafik akurasi ZCR dan CNN, terdapat peluang bahwa semakin panjang durasi maka akurasi semakin baik, tetapi tidak menutup kemungkinan akurasi dapat semakin menurun. Terlihat bahwa untuk durasi 9 detik, akurasi dapat mencapai 61 %, setelah dicoba untuk durasi 10, 11, 12, 13, dan 14 detik, akurasi justru menurun, tetapi pada durasi 15 detik akurasi dapat mengalami kenaikan lagi hingga 62 %, oleh karena itu dapat disimpulkan bahwa semakin panjang durasi memiliki peluang untuk mencapai akurasi yang tinggi, tetapi juga tidak menutup kemungkinan menemui akurasi yang semakin menurun.

Gambar 4. 17 Akurasi *Zero Crossing Rate* dan CNN

Grafik *loss function* fitur ZCR dan CNN diperlihatkan pada Gambar 4. 18, nilai loss masih konsisten di atas 1.23 dan di bawah 1.39, nilai ini tidak memberikan perubahan yang signifikan terhadap perubahan durasi.

Gambar 4. 18 *Loss Zero Crossing Rate*

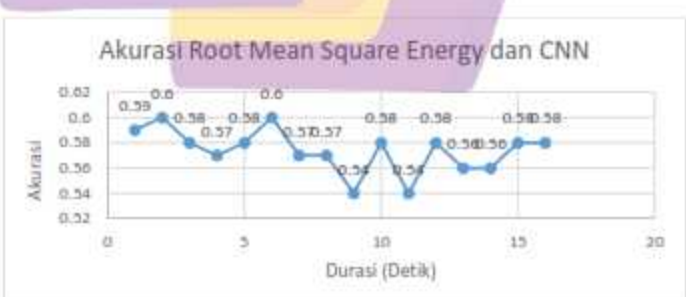
Hasil evaluasi model dengan fitur *Energy* memperoleh akurasi tertinggi pada durasi 2 dan 6 detik yaitu 60 %, hasil ujicoba *Energy* dan CNN dari 1 sampai dengan 16 detik ditunjukkan pada Tabel 4. 5.

Tabel 4. 5 Ujicoba *Energy* dan CNN

Durasi (detik)	Proses Penentuan Bentuk	Bentuk	Akurasi	Loss
1	$1\,x\,\frac{22050}{512}$	13, 44, 1	0.59	1.20
2	$2\,x\,\frac{22050}{512}$	13, 87, 1	0.6	1.25
3	$3\,x\,\frac{22050}{512}$	13, 130, 1	0.58	1.32
4	$4\,x\,\frac{22050}{512}$	13, 173, 1	0.57	1.3
5	$5\,x\,\frac{22050}{512}$	13, 216, 1	0.58	1.29

Tabel 4. 5 Ujicoba *Energy* dan CNN (Lanjutan)

Durasi (detik)	Proses Penentuan Bentuk	Bentuk	Akurasi	Loss
6	$6\,x\,\frac{22050}{512}$	13, 259, 1	0.6	1.21
7	$7\,x\,\frac{22050}{512}$	13, 302, 1	0.57	1.33
8	$8\,x\,\frac{22050}{512}$	13, 345, 1	0.57	1.29
9	$9\,x\,\frac{22050}{512}$	13, 388, 1	0.54	1.39
10	$10\,x\,\frac{22050}{512}$	13, 431, 1	0.58	1.29
11	$11\,x\,\frac{22050}{512}$	13, 474, 1	0.54	1.35
12	$12\,x\,\frac{22050}{512}$	13, 517, 1	0.58	1.31
13	$13\,x\,\frac{22050}{512}$	13, 560, 1	0.56	1.29
14	$14\,x\,\frac{22050}{512}$	13, 603, 1	0.56	1.28
15	$15\,x\,\frac{22050}{512}$	13, 645, 1	0.58	1.25
16	$16\,x\,\frac{22050}{512}$	13, 690, 1	0.58	1.3



Gambar 4. 19 Akurasi *Energy* dan CNN

Berdasarkan grafik perolehan akurasi fitur *Energy* dan CNN yang ditunjukkan pada Gambar 4. 19 dapat disimpulkan bahwa akurasi pada durasi yang pendek (1,2, dan 3 detik) telah menunjukkan akurasi yang cukup tinggi (59 %, 60 %, dan 58 %), tetapi setelah dilakukan percobaan dengan durasi yang semakin panjang, justru akurasi semakin menurun. Oleh karena itu, dapat disimpulkan bahwa fitur *Energy* dapat mencapai akurasi tertingginya dengan durasi yang lebih pendek daripada ZCR.

Sedangkan perolehan *loss* dari fitur *Energy* dan CNN yang ditunjukkan pada Gambar 4. 20 masih konsisten di atas 1.2, perolehan loss *Energy* masih di angka yang sama dengan *loss* yang diperoleh ZCR.

Gambar 4. 20 Loss *Energy* dan CNN

4.4. Confusion Matrix

Confusion matrix digunakan untuk mengukur kinerja dari sebuah model yang telah dibuat. Library *sklearn.metrics import confusion_matrix* dan *sklearn.metrics import classification_report* digunakan untuk menghitung nilai *confusion matrix*. Gambar 4. 21 merupakan perulangan untuk memprediksi seluruh

data testing yang berjumlah 256 data, fungsi perulangan ini menghasilkan array *y_true* (label sebenarnya) dan array *y_prediksi* (label hasil prediksi) yang kemudian akan digunakan sebagai parameter *confusion_matrix*.

```
y_prediksi = []
y_target = []
ulang = 256

for i in range(ulang):
    print('prediksi ke-', str(i))
    X_to_predict = X_test[i]
    y_to_predict = y_test[i]
    predict(model, X_to_predict, y_to_predict)
    y_target.append(y_to_predict)
    print('array y_true', y_target)

print('array y_prediksi', y_prediksi)
```

Gambar 4. 21 Perulangan untuk Memprediksi 256 Data Testing

Gambar 4. 22 merupakan fungsi untuk menghitung *confusion_matrix*, parameter yang diperlukan adalah *y_true* (label sebenarnya) dan *y_pred* (label hasil prediksi) yang dihasilkan dari fungsi perulangan. Fungsi *confusion_matrix (y_true, y_pred)* digunakan untuk menghitung nilai *confusion matrix* seperti terlihat pada Gambar 4. 23, sedangkan fungsi *classification_report (y_true, y_pred, target_names=target_names)* digunakan untuk mendapatkan nilai *precision, recall,* dan *f1-score* dari tiap kelas serta nilai keseluruhan *accuracy*, seperti Gambar 4. 24.

```
#y_true = label sebenarnya
y_true = [1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 5, 2, 5, 5, 1, 3, 3, 1, 5, 2, 4,
#y_pred = hasil prediksi
y_pred = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 5, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1,
f, ax = plt.subplots(figsize=(8,5))
sns.heatmap(confusion_matrix(y_true, y_pred), annot=True, fmt=".0f", ax=ax)
plt.xlabel("y_pred")
plt.ylabel("y_true")
plt.show()
target_names = ['arabic', 'english', 'french', 'korean', 'mandarin', 'spanish']
print(classification_report(y_true, y_pred, target_names=target_names))
```

Gambar 4. 22 Fungsi *Confusion Matrix* untuk fitur ZCR

Gambar 4. 23 Nilai *Confusion Matrix*

	precision	recall	f1-score	support
arabic	0.50	0.15	0.24	26
english	0.77	0.89	0.88	166
french	0.20	0.09	0.13	11
korean	0.00	0.00	0.00	10
mandarin	0.20	0.17	0.18	12
spanish	0.17	0.16	0.17	31
accuracy			0.62	256
macro avg	0.30	0.24	0.25	256
weighted avg	0.56	0.62	0.58	256

Gambar 4. 24 Hasil Perhitungan *Accuracy, Precision, Recall,* dan *F1-Score*

Nilai *precision*, *recall*, dan *F1-Score* dihitung pada setiap durasi (1-16 detik), berdasarkan hasil perhitungan *precision (P), recall (R),* dan *F1-Score (F)* fitur ZCR yang ditunjukkan pada Tabel 4. 6 dapat dilihat bahwa hanya kelas *English* yang memiliki nilai P, R dan F dari 1 hingga 16 detik, sedangkan kelas lainnya berangsur naik dari 0 hingga nilai tertentu (dari 1 hingga 16 detik), adanya nilai 0 dikarenakan nilai *confusion matrix y_pred* (*x axis*) dan *y_true* (*y axis*) bernilai 0 atau dengan kata lain label sebenarnya tidak dapat diprediksi. Misal, Gambar 4. 23 *confusion matrix* dengan *y_pred* berlabel 3 dan *y_true* berlabel 3 menunjukkan nilai 0, hal ini menunjukkan kelas dengan index 3 (*Korean*) tidak dapat diprediksi, sehingga nilai P, R, dan F bernilai 0. Perubahan nilai P, R, dan F pada setiap kelas dan durasi ditunjukkan pada Gambar 4. 25.

Nilai *precision*, *recall*, dan *F1-Score* yang ditunjukkan pada Gambar 4. 25 memperlihatkan bahwa semakin panjang durasi maka kelas yang diprediksi semakin banyak, seperti durasi 1-2 detik hanya dapat memprediksi 1 kelas yaitu *English*, 2-4 detik dapat memprediksi 2 kelas yaitu *English* dan *Spanish*, 5-12 detik dapat memprediksi 3 kelas yaitu *Arabic, English, Spanish*, dan 15 detik dapat memprediksi 5 kelas yaitu *Arabic, English, French, Mandarin, Spanish*, serta 16 detik dapat memprediksi *Arabic, English, Korean, Mandarin, dan Spanish*. Percobaan ini membuktikan bahwa fitur ZCR bekerja lebih baik pada durasi yang panjang, artinya semakin panjang durasi maka semakin banyak kelas yang dapat diprediksi.

Nilai *precision, recall,* dan *F1-Score* fitur *Energy* ditunjukkan pada Tabel 4. 7, sama halnya dengan fitur ZCR, pada fitur Energy hanya kelas English yang memiliki nilai P, R dan F dari 1 hingga 16 detik, sedangkan kelas lainnya berangsur naik dari 0 hingga nilai tertentu (dari 1 hingga 16 detik), adanya nilai 0 dikarenakan nilai *confusion matrix y_pred* (*x axis*) dan *y_true* (*y axis*) bernilai 0 atau dengan kata lain label sebenarnya tidak dapat diprediksi. Perubahan nilai P, R, dan F pada setiap kelas dan durasi ditunjukkan pada Gambar 4. 26.

Gambar 4. 26 memperlihatkan bahwa Fitur *Energy* hanya dapat memprediksi 2 kelas yaitu *English* dan *Spanish* dari 1 hingga 16 detik, sedangkan kelas lainnya seperti *Arabic, French, Korean*, dan *Mandarin* tidak berhasil diprediksi (P, R, dan F bernilai 0). Percobaan ini membuktikan bahwa *Energy* bekerja pada durasi yang singkat dan hanya mampu memprediksi kelas yang sedikit.

Tabel 4. 6 Nilai *Precision (P)*, *Recall (R)*, dan *F1-Score (F)* Fitur ZCR

	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Durasi (Detik)	1			2			3			4			5			6			7			8		
Arabic	0	0	0	0	0	0	0	0	0	0	0	0	0.23	0.13	0.17	0.5	0.03	0.06	0.33	0.07	0.11	0.14	0.05	0.07
English	0.57	1	0.73	0.59	1	0.74	0.57	1	0.72	0.59	0.99	0.74	0.61	0.97	0.74	0.62	0.98	0.76	0.66	0.93	0.77	0.63	0.96	0.76
French	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Korean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mandarin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spanish	0	0	0	0	0	0	0.5	0.03	0.05	0.3	0.03	0.06	0.43	0.07	0.12	0.27	0.15	0.19	0.22	0.22	0.22	0.24	0.13	0.17

	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Durasi (Detik)	9			10			11			12			13			14			15			16		
Arabic	0.12	0	0.07	0.23	0.1	0.15	0.33	0.07	0.12	0.12	0.05	0.07	0	0	0	0	0	0	0.5	0.15	0.24	0.38	0.16	0.22
English	0.64	1	0.77	0.66	0.9	0.77	0.63	0.96	0.76	0.68	0.89	0.77	0.65	0.95	0.77	0.68	0.83	0.74	0.73	0.89	0.8	0.64	0.87	0.74
French	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.09	0.13	0	0	0
Korean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.09	0.13
Mandarin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0.08	0.07	0.2	0.17	0.18	0.17	0.1	0.12
Spanish	0.4	0	0.23	0.27	0.3	0.26	0.33	0.26	0.29	0.22	0.22	0.22	0.17	0.11	0.13	0.37	0.27	0.31	0.17	0.16	0.17	0.42	0.33	0.37

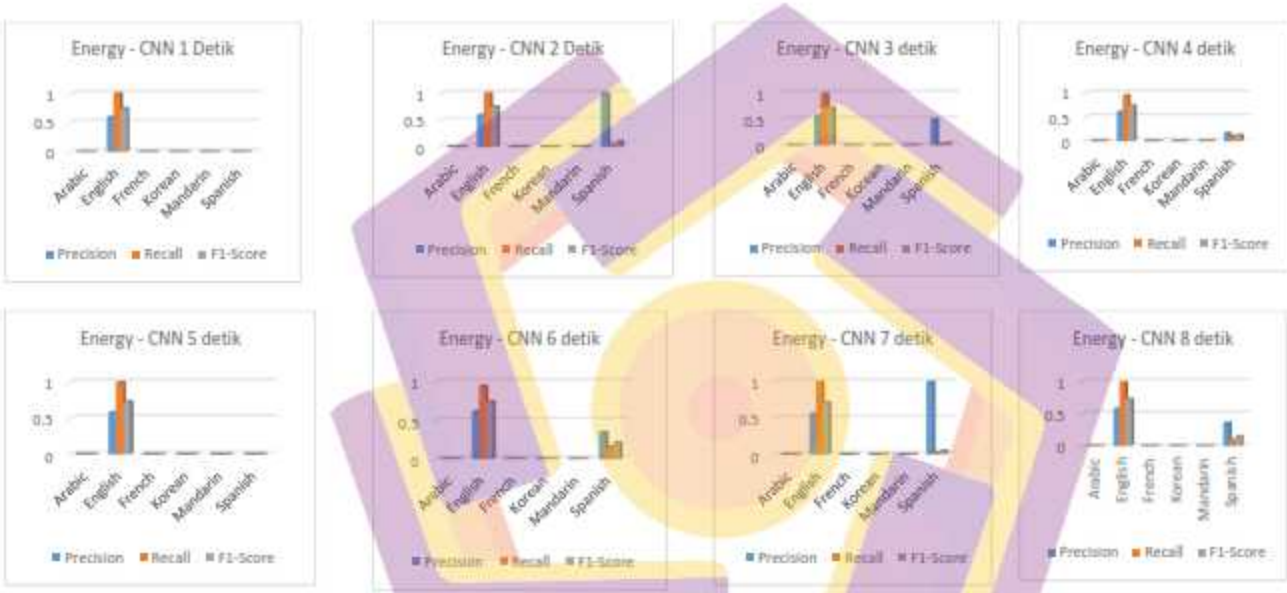
Gambar 4. 25 Grafik Nilai *Precision*, *Recall*, *F1-Score* Fitur ZCR

Gambar 4.25 Grafik Nilai *Precision*, *Recall*, *F1-Score* Fitur ZCR (Lanjutan)

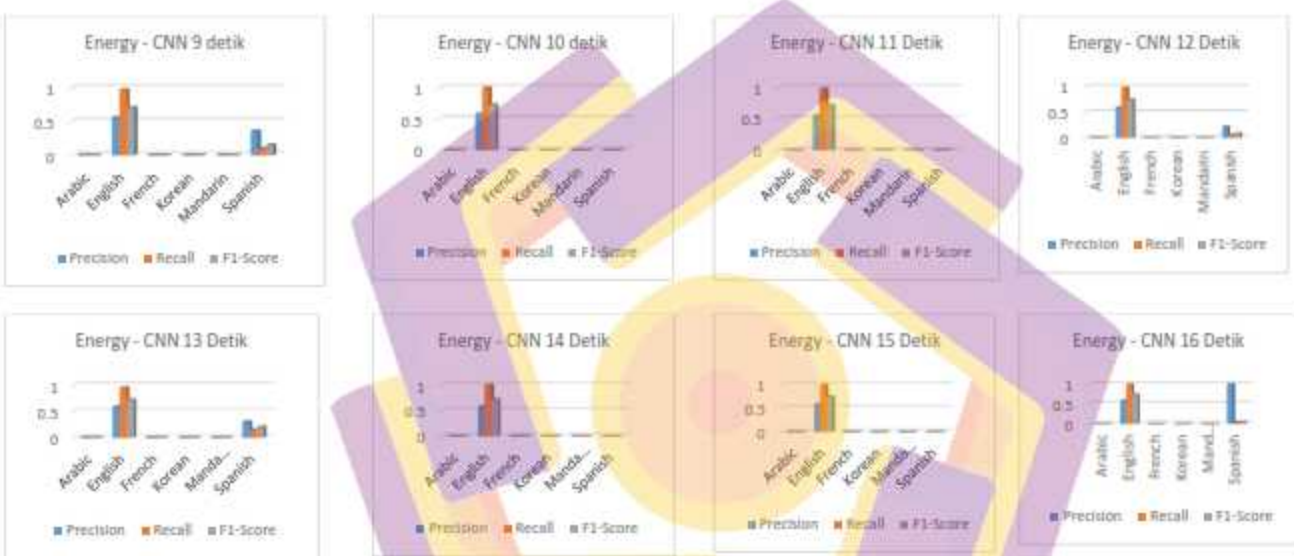
Tabel 4. 7 Nilai *Precision* (P), *Recall* (R), dan *F1-Score* (F) Fitur *Energy*

	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Durasi (Detik)	1			2			3			4			5			6			7			8		
Arabic	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
English	0.59	1	0.75	0.59	1	0.75	0.58	0.99	0.73	0.61	0.95	0.74	0.58	1	0.73	0.62	0.95	0.75	0.56	1	0.72	0.58	0.99	0.73
French	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Korean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mandarin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spanish	0	0	0	1	0.05	0.1	0.5	0.03	0.05	0.17	0.1	0.13	0	0	0	0.35	0.16	0.22	1	0.03	0.05	0.36	0.1	0.15

	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Durasi (Detik)	9			10			11			12			13			14			15			16		
Arabic	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
English	0.56	0.97	0.71	0.58	1	0.73	0.55	0.98	0.71	0.6	0.99	0.75	0.58	0.95	0.72	0.57	0.99	0.72	0.59	0.99	0.74	0.59	0.99	0.74
French	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Korean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mandarin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Spanish	0.36	0.1	0.15	0	0	0	0	0	0	0.22	0.06	0.09	0.3	0.15	0.2	0	0	0	0	0	0	1	0.03	0.05



Gambar 4. 26 Grafik Nilai *Precision*, *Recall*, dan *F1-Score* Fitur *Energy*



Gambar 4.26 Grafik Nilai *Precision*, *Recall*, dan *F1-Score* Fitur *Energy* (Lanjutan)

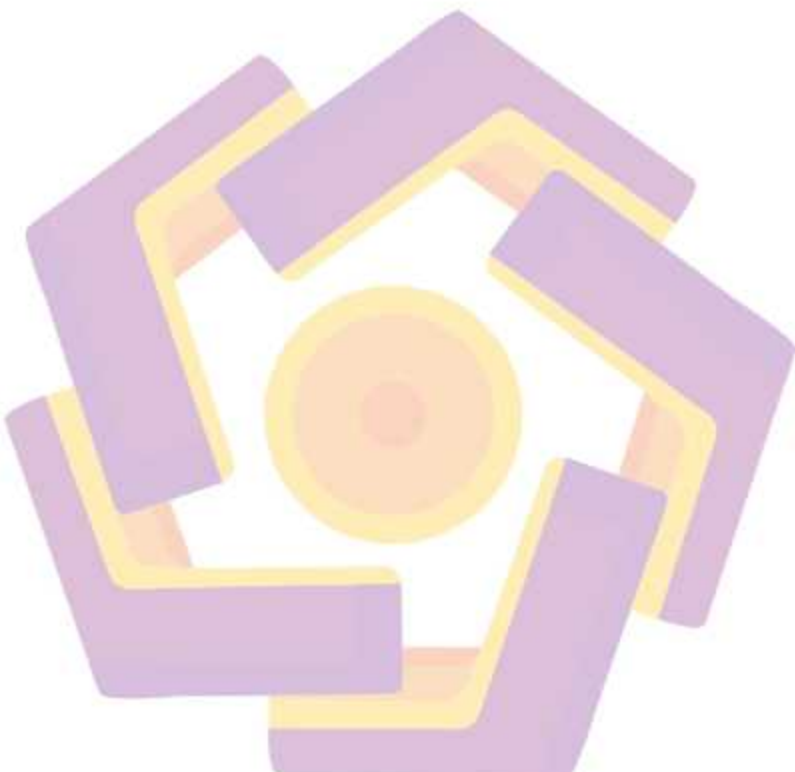
Berdasarkan percobaan yang telah dilakukan, dapat dilihat bahwa semakin panjang durasi, fitur ZCR memiliki peluang untuk mencapai akurasi yang tinggi. Sedangkan, fitur *Energy* tidak menunjukkan adanya kenaikan akurasi pada durasi yang panjang. *Energy* dapat mencapai akurasi pada durasi yang singkat yaitu 60 % (2 detik), sedangkan fitur ZCR mencapai akurasi tertinggi pada durasi yang panjang yaitu 62 % (15 detik).

Berdasarkan evaluasi model menggunakan *confusion matrix*, dapat dilihat bahwa *Energy* hanya dapat melakukan klasifikasi pada 2 kelas yaitu *English* dan *Spanish*, sedangkan kelas lainnya tidak pernah berhasil diklasifikasi walaupun dengan durasi yang panjang. Hal ini menunjukkan, walaupun mampu mencapai akurasi yang tinggi, tetapi *Energy* tidak dapat mewakili semua kelas. Sedangkan, fitur ZCR mampu mencapai akurasi yang tinggi walaupun harus melalui durasi yang lebih panjang, fitur ZCR mampu mengklasifikasikan hingga 5 kelas sampai dengan durasi 16 detik, sehingga berdasarkan hasil *confusion matrix* fitur ZCR dapat dikatakan lebih baik karena dapat mewakili hampir semua kelas.

English, Arabic, Spanish, dan *Mandarin* merupakan kelas yang paling sering berhasil dilakukan klasifikasi, sedangkan *French* dan *Korean* seringkali gagal dalam klasifikasi. Hal ini mungkin dikarenakan jumlah dataset terkecil dimiliki oleh *French* dan *Korean*, pada dasarnya kelas *French* dan *Korean* dapat dilakukan klasifikasi jika durasi lebih panjang walaupun dengan jumlah dataset yang sedikit.

Penelitian ini memberikan kontribusi di bidang *Speech Recognition* agar menjadi lebih canggih. Dibandingkan dengan penelitian terdahulu, akurasi yang

diberikan dapat lebih baik dengan menggunakan Fitur *Energy* dan ZCR. Selain itu, penelitian ini memberikan kontribusi pada ide pengolahan fitur menjadi 2 dimensi. Pada dasarnya, Ekstraksi Fitur audio menghasilkan array 1 dimensi, sehingga apabila menggunakan klasifikasi CNN 2D, array tersebut harus diolah menjadi 2 dimensi atau lebih.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian, dapat diambil kesimpulan bahwa:

1. Tingkat akurasi tertinggi yang didapatkan dari fitur *Zero Crossing Rate* (ZCR) dan CNN yaitu 62 % dan loss 1.32 untuk durasi 15 detik.
2. Tingkat akurasi tertinggi yang didapatkan dari fitur *Energy* dan CNN yaitu 60 % dan loss 1.21 untuk durasi 2 dan 6 detik.
3. Berdasarkan hasil percobaan, fitur audio terbaik yaitu ZCR dengan akurasi tertinggi 62 %, dan dapat memprediksi hingga 5 kelas dalam durasi 15 detik.
4. Tingkat akurasi yang diperoleh dari percobaan 6 kelas lebih baik dibandingkan dengan penelitian sebelumnya yaitu di atas 53 %.
5. Kenaikan akurasi dari penelitian sebelumnya sebesar 21, 57 %.

5.2. Saran

Penelitian ini dapat dikembangkan dengan memperhatikan hal berikut:

1. Hasil ekstraksi fitur audio sebenarnya merupakan array 1 dimensi (44), sehingga diperlukan pemrosesan data fitur untuk membuat fitur tersebut menjadi array 2 dimensi, percobaan kali ini yaitu membuat fitur menjadi 2 dimensi dengan cara menduplikasi kolom menjadi 13 baris (13, 44, 1). Percobaan berikutnya dapat menemukan ide pemrosesan data fitur sehingga menjadi array 3 dimensi dan menjadi *input* CNN.

2. Semakin panjang durasi, maka fitur ZCR memiliki peluang mendapatkan akurasi yang lebih baik, oleh karena itu percobaan berikutnya dapat meneliti durasi di atas 16 detik.
3. Percobaan ini hanya dapat memproses maksimal berdurasi 16 detik, ketika dilakukan percobaan pada durasi 17 sampai 52 detik tidak dapat dilakukan klasifikasi karena tidak adanya keseragaman durasi. Percobaan berikutnya dapat mempersiapkan dataset agar dapat dilakukan ekstraksi fitur hingga 52 detik.
4. *French* dan *Korean* merupakan aksen yang paling sulit diprediksi, percobaan berikutnya dapat menemukan sebuah model agar kelas tersebut berhasil diprediksi.

DAFTAR PUSTAKA

Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Jurnal Teknologi Indonesia*.

Al-Azhar, & Nuh, M. (2011). *Audio forensic: Theory and analysis*. Pusat Laboratorium Forensik Polri Bidang Fisika Dan Komputer Forensik.

Bachu, R. G., Kopparthi, S., Adapa, B., & Barkana, B. D. (2010). Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy. *Advanced Techniques in Computing Sciences and Software Engineering*, 279–282. https://doi.org/10.1007/978-90-481-3660-5

Barkana, B. D., & Patel, A. (2020). Analysis of vowel production in Mandarin/Hindi/American- accented English for accent recognition systems. *Applied Acoustics*, *162*, 107203. https://doi.org/10.1016/j.apacoust.2019.107203

Bryant, M., Chow, A., & Li, S. (2014). *Classification of Accents of English Speakers by Native Language*. 1–5. http://cs229.stanford.edu/proj2014/Morgan Bryant, Amanda Chow, Sydney Li, Classification of Accents of English Speakers by Native Language.pdf

Chionh, K., Song, M., & Yin, Y. (2018). *Application of Convolutional Neural Networks in Accent Identification*.

Giannakopoulos, Theodore, Spyrou, E., & Perantonis, S. J. (2019). Recognition of urban sound events using deep context-aware feature extractors and handcrafted features. In *IFIP Advances in Information and Communication Technology* (Vol. 560). Springer International Publishing. https://doi.org/10.1007/978-3-030-19909-8_16

Giannakopoulos, Theodoros, & Pikrakis, A. (2014). Audio Features. In *Introduction to Audio Analysis* (pp. 59–103). Elsevier. https://doi.org/10.1016/B978-0-08-099388-1.00004-2

Hariyani, Y. S., Hadiyoso, S., & Siadari, T. S. (2020). Deteksi Penyakit Covid-19 Berdasarkan Citra X-Ray Menggunakan Deep Residual Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *8*(2), 443. https://doi.org/10.26760/elkomika.v8i2.443

Haykin, S. (2009). Neural Networks and Learning Machines. In *Pearson Prentice Hall*. https://doi.org/10.1016/B978-0-12-809633-8.20339-7

Hernandez, S. P., Bulitko, V., Carleton, S., Ensslin, A., & Goorimoorthee, T. (2018). *Deep Learning for Classification of Speech Accents in Video Games*.

Hidayat, T., Zakaria, M. H., & Pee, A. N. C. (2018). Comparison of lossless compression schemes for WAV audio data 16-bit between huffman and coding arithmetic. *International Journal of Simulation: Systems, Science and Technology*, *19*(6), 36.1-36.7. https://doi.org/10.5013/IJSSST.a.19.06.36

Honnavalli, D., & Shylaja, S. S. (2019). *Supervised Machine Learning Model for Accent Recognition in English Speech using Sequential MFCC Features*.

Jondya, A. G., & Iswanto, B. H. (2018). Analisis dan Seleksi Fitur Audio pada Musik Tradisional Indonesia. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, *4*(2), 77.

https://doi.org/10.24014/coreit.v4i2.6506

Kim, P. (2017). MATLAB Deep Learning. *MATLAB Deep Learning*, *November 2013*, 121–147. https://doi.org/10.1007/978-1-4842-2845-6

Knees, P., & Schedl, M. (2016). *Basic Methods of Audio Signal Processing*. https://doi.org/10.1007/978-3-662-49722-7_2

Manliguez, C. (2016). *Generalized Confusion Matrix for Multiple Classes*. *November*, 5–7. https://doi.org/10.13140/RG.2.2.31150.51523

Mcfee, B., Raffel, C., Liang, D., Ellis, D. P. W., Mcvicar, M., Battenberg, E., & Nieto, O. (2015). Librosa - audio processing Python library. *Proc. Of The 14Th Python in Science Conf, Scipy*, 18–25. http://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcfee.pdf

Nilam, H., Adiarso, P., Osmond, A. B., Elektro, F. T., Telkom, U., Jember, J., Recognition, S., & Network, R. N. (2019). *Penentuan Dialek Jawa Menggunakan Metode Recurrent Neural Network Determination of Java Dialek Using Recurrent Neural Network Method*. *6*(2), 5625–5636.

Nurhikmat, T. (2018). Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Citra Wayang Golek. *Tugas Akhir*.

Patel, A., & Barkana, B. D. (2018). Analysis of American English corner vowels produced by Mandarin, Hindi, and American accented speakers and a baseline accent recognition system. *2018 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2018*, 1–5. https://doi.org/10.1109/LISAT.2018.8378031

Perdana, R. R., Fatichah, C., & Soelaiman, R. (2017). Implementasi Ekstraksi Fitur untuk Pengelompokan Berkas Musik Berdasarkan Kemiripan Karakteristik Suara. *Jurnal Teknik ITS*, *6*(1). https://doi.org/10.12962/j23373539.v6i1.22138

Singh, Y., Pillay, A., & Jembere, E. (2019). *Features of speech audio for deep learning accent recognition*. 4–6.

Singh, Y., Pillay, A., & Jembere, E. (2020). Features of speech audio for accent recognition. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, IcABCD 2020 - Proceedings*. https://doi.org/10.1109/icABCD49160.2020.9183893

Viranda, R. S. (2017). *Ekstraksi Ciri dan Identifikasi Sinyal Suara Jantung S1 dan S2 Phonocardiogram (PCG) Menggunakan Metode Wavelet Packet Transform*.

Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, *507*, 772–794. https://doi.org/10.1016/j.ins.2019.06.064