Deteksi Ucapan untuk Sistem Pengawasan Asesmen (iProctor) Menggunakan Metode *Deep Learning*

Muhammad Iqbal Izzul Haq, Dini Adni Navastara, dan Shintami Chusnul Hidayati
Departemen Informartika , Institut Teknologi Sepuluh Nopember (ITS)
e-mail: dini_navastara@if.its.ac.id

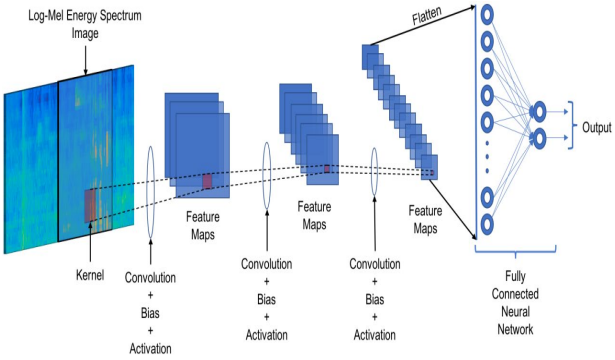
***Abstrak*—Asesmen adalah kegiatan mengumpulkan informasi ketercapaian kompetensi siswa. Asesmen merupakan bagian integral dari proses pembelajaran, tak terkecuali dalam pembelajaran berbasis Out Door Learning (ODL) dan Massive Open Online Course (MOOC). Sebuah studi menyatakan bahwa persentase siswa yang melakukan kecurangan dalam pelaksanaan kegiatan akademik terus meningkat, dan lebih mudah bagi mereka untuk berlaku curang pada asesmen yang dilakukan secara daring. Hal ini menjadi tantangan untuk perkembangan iProctor, yaitu platform untuk melakukan asesmen seara daring. Untuk mengurangi risiko kecurangan, sistem pelaksanaan dan pengawasan ujian yang valid menjadi suatu hal yang penting. Pada penelitian ini diuji sistem pengawasan otomatis bedasarkan audio. Data audio didapatkan dari mikrofon yang terletak pada ruang dilakukannya asesmen. Sistem pengawasan asesmen dilakukan secara otomatis dengan metode deteksi ucapan menggunakan metode deep learning dengan model CNN. Data audio di ekstraksi fitur menggunakan log-mel spectrogram. Hasil esktraksi fitur menjadi input model CNN MobileNetV3. Hasil prediksi dari MobileNetV3 dilakukan proses smoothing dengan metode Majority Vote. Hasil penelitian ini menunjukkan bahwa model deteksi ucapan memberikan hasil terbaik dengan model CNN MobileNetV3-Large pada dataset librispeech dengan speech f1 score 0.8652, non-speech f1 score 0.7332, dan hasil weighted average 0.8242. Ekstraksi fitur menggunakan metode log-mel spectrogram menggunakan parameter fft size 512, mel bins 40, hope size 8, lower frequency 300, upper frequency 8000. Hasil dari log-mel spectrogram dibagi menjadi banyak frame 25ms dan step 12.5ms atau overlap 50.**

***Kata Kunci*—*Deep Learning*, Deteksi Ucapan, iProctor, *Log-mel spectrogram*, MobileNetV3.**

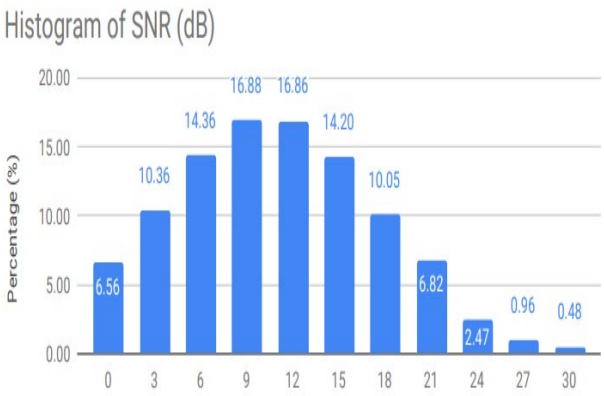
I. PENDAHULUAN

SALAH satu perilaku curang yang memungkinkan terjadi dalam pelaksanaan asesmen secara daring adalah tindakan peserta yang mengajukan pertanyaan secara verbal. Oleh karena itu, diusulkan pemanfaatan algoritma pendeteksi ucapan berdasarkan data suara. Dalam beberapa tahun terakhir banyak pendekatan yang telah dicoba untuk menyelesaikan masalah deteksi ucapan dengan algoritma *VAD* atau *Voice Activity Detection* [1].

Metode deteksi ucapan menggunakan *deep learning* seperti yang dilakukan pada penelitian menggunakan ekstraksi fitur *log-mel spectrogram*, yang terbukti memiliki performa tinggi dibanding metode ekstraksi fitur lainnya menggunakan spektrogram [2]. Berdasarkan hal-hal di atas, pada tugas akhir ini, diusulkan otomatisasi pengawasan asesmen melalui pemanfaatan Deep Learning Arsitektur *Convolutional Neural Network (CNN)* untuk deteksi ucapan (*Voice Activity Detection, VAD*). *Log-mel spectrogram* digunakan sebagai metode ekstraksi fitur untuk input dari Model CNN.



Gambar 1. Ekstraksi Fitur *log-mel spektrogram*.



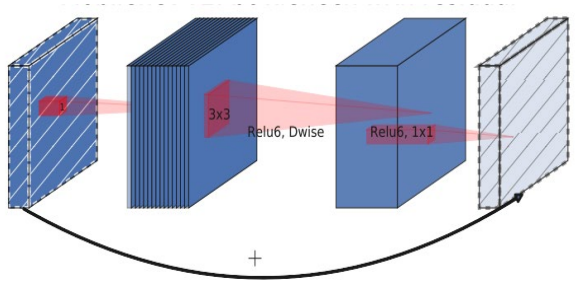
Gambar 2. Distribusi SNR.

II. TINJAUAN PUSTAKA

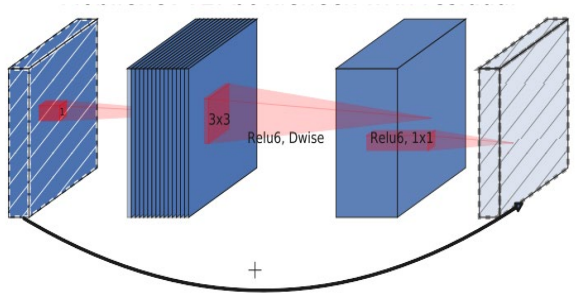
A. Hasil Penelitian Terdahulu

1) CNN VAD

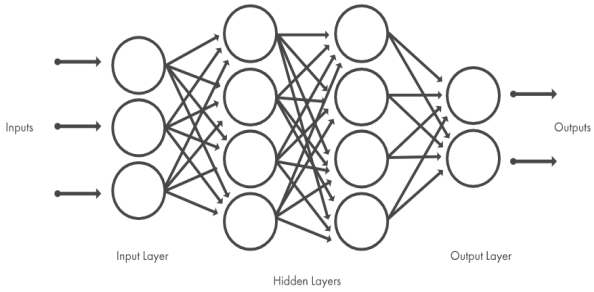
Penelitian ini juga membandingkan beberapa metode deteksi ucapan yang menggunakan algoritma dan arsitektur model deep learning. Berbagai macam metode deteksi ucapan dievaluasi performanya dalam deteksi ucapan pada berbagai kondisi kebisingan. Input dari model CNN haruslah berupa gambar maka data audio diubah menjadi data gambar terlebih dahulu. Metode yang dipilih oleh penelitian Sehgal menggunakan *log-mel filterbank energy*. Masalah selanjutnya yang diatasi oleh penelitian ini adalah tingkat keterlambatan waktu dari metode *deep learning* yang tinggi. Penelitian oleh Sehgal mengurangi tingkat keterlambatan waktu dengan mencari potongan data audio yang paling sesuai untuk diambil (*window size* 25ms, *offset* 12.5ms), mengoptimalkan parameter untuk *log-mel* spektrogram (*fft size* 512, *log mel bins* 40), dan menringkaskan arsitektur CNN yang digunakan seperti yang ditunjukkan Gambar 1 [2].



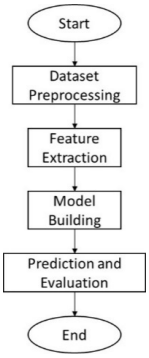
Gambar 3. Bottleneck Layer MobileNetV3.



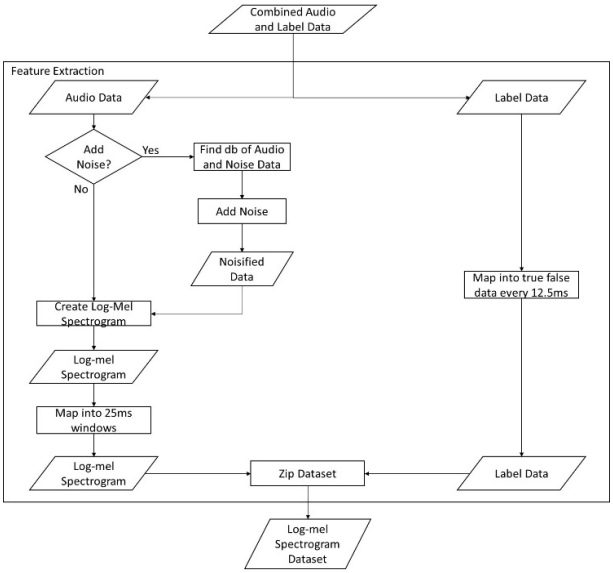
Gambar 4. Bottleneck Layer MobileNetV3 SE.



Gambar 5. Diagram CNN.



Gambar 6.Bagan Perancangan Sistem Keseluruhan.



Gambar 7.Bagan Model Sistem.

2) Librispeech Dataset

Dataset yang digunakan oleh penelitian dalam *training* model CNN mereka merupakan data audio dari *librispeech*. Karena dataset dari *librispeech* sudah memiliki data ucapan dengan kebisingan dan tanpa kebisingan sebanyak 960 jam. Namun, data audio dari dataset *librispeech* tidak memiliki data label yang sesuai untuk kebtutuhan *training*. Data label dari *librispeech* hanya menyediakan kata yang diucapkan oleh data audio tanpa adanya waktu kapan kata tersebut diucapkan. Maka sebelum dataset digunakan dalam *training* model CNN, data label diproses melalui *forced alignment* oleh *speech recognition* model.

3) Signal to Noise Ratio

Pada penelitian ini dibuat metode simulasi ruangan, simulasi ruangan ini menghasilkan dari data audio yang memiliki tingkat kebisingan rendah dapat diproses dan ditambah tingkat kebisingannya dengan cara mensimulasikan data audio bersih pada ruangan yang acak. Metode penambahan kebisingan ini menghasilkan distribusi kebisingan dengan SNR (Signal to Noise Ratio) 0db yang paling bising, SNR 30db yang paling sunyi, dan terbanyak data audio memiliki SNR 9 sampai 12 db. Distribusi SNR digunakan sebagai basis dari penambahan kebisingan secara *adaptive* pada penelitian ini seperti yang ditunjukkan Gambar 2 [3].

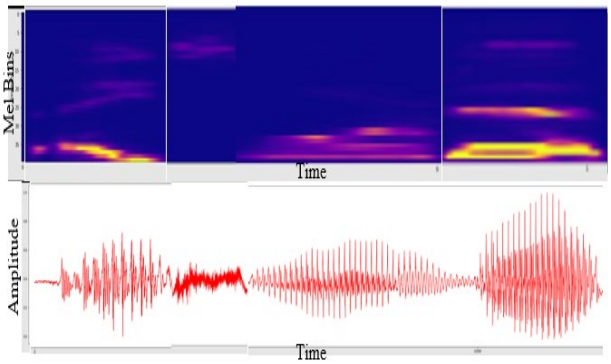
4) Performa Spektrogram

Pada penelitian ini, beberapa metode untuk ekstraksi fitur data audio menjadi data input model CNN dibandingkan performanya dalam klasifikasi beberapa jenis kebisingan yang mungkin terjadi dalam kehidupan sehari-hari, contohnya adalah ac pendingin ruangan, suara mesin berat, anak kecil, anjing menggonggong, suara klakson mobil, dan masih banyak lagi. Metode ekstraksi fitur data audio yang digunakan antara lainnya, *Mel-frequency cepstral coefficients* (MFCC), *continuous Wavelet transform* (CWT), *constant-Q transform* (CQT), *Mel-scaled STFT, linear-scaled STFT*. Metode ekstraksi spektral atau 5 metode ekstraksi fitur yang disebutkan diatas diuji dan dibandingkan hasilnya. Penelitian ini menghasilkan bahwa, dari 10 kelas, hasil dari *linear-STFT, Mel-STFT* dan *CQT* memiliki performa yang lebih baik. Selain itu, ditemukan juga performa dari *CWT* dan *MFCC* memiliki kemiripan yang tinggi, kemiripan dari bukan hanya dari segi akurasi, namun dalam 10 kelas yang perlu diklasifikasi dalam pengujian ini, metode *CWT* dan *MFCC* memiliki kesalahan yang serupa. Setelah itu metode Mel-STFT, Linear-STFT, dan CQT diuji lebih lanjut, dihasilkan Mel-STFT memiliki performa yang terbaik dalam semua variasi yang diuji, meskipun Linear-STFT dan CQT memiliki performa yang baik dalam beberapa model, namun CWT dan MFCC memiliki performa paling rendah dibanding dengan 3 metode lain yang diuji [4].

B. Dasar Teori

1) MobileNetV3

Model CNN ini dibuat khususnya untuk mencari model CNN yang memiliki performa tinggi sedangkan bisa



Gambar 8.Hasil *Log-mel Spectrogram.*

Tabel 1.
Arsitektur Model CNN (Sehgal).

Layer	Nodes
conv2d, 5x5	40
conv2d, 5x5	20
conv2d, 5x5	10
Fully Connected	100
softmax	2

Tabel 2.
Hardware yang digunakan.

Komponen	Spesifikasi
OS	Ubuntu 18.04.3 LTS
CPU	Intel(R) Xeon(R) 2.30GHz
RAM	12GB
VGA	Nvidia K80

dijalankan dalam mesin berspesifikasi rendah. Struktur dari MobileNetV3 terdiri dari *layer bottleneck* yang menjadi kunci dari efesiensi model CNN ini. *Layer bottleneck* terdiri dari 1 *expansion layer CNN* 1x1, 1 *layer depthwise convolution, 1 squeeze layer CNN 1x1 16 node*.

Expansion layer memiliki jumlah *filters* yang lebih besar dari pada output, semisal 384 filters, lalu semua layer tersebut dilakukan *depthwise convolution,* dan akhirnya melewati *squeeze layer* yang memiliki jumlah *filters* yang jauh lebih kecil dari *expansion layer* semisal 64 nodes. Pada contoh ini *bottleneck layer* memiliki *expansion factor* sebesar 6 yang didapat oleh *expansion layer* 384 dibagi dengan *squeeze layer* 64.

Penemuan selanjutnya dari MobileNetV3 menggunakan *squeeze and excite* block pada proses sebelum *squeeze layer* dengan tujuan untuk mengurangi informasi yang hilang pada *squeeze layer. Squeeze and excite layer* mengurangi informasi yang hilang dengan meningkatkan performa pada model, terutama pada model yang sangat besar seperti yang ditunjukkan Gambar 3 dan Gambar 4 [3], [5].

2) Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu algoritma paling populer pada deep learning, yang sangat berguna untuk menemukan pola dalam gambar, teks, suara, hingga video untuk mengenali objek, wajah. CNN belajar langsung dari data, menggunakan pola untuk mengklasifikasikan dan menghilangkan kebutuhan untuk ekstraksi fitur secara manual. CNN dapat memiliki puluhan bahkan ratusan layer yang masing-masing belajar untuk mendeteksi fitur gambar yang berbeda. Filter diterapkan ke setiap gambar training pada resolusi yang berbeda dan output dari masing-masing gambar digunakan sebagai input ke layer berikutnya. Seperti *neural network* lainnya, CNN terdiri dari

Tabel 3.
Software yang digunakan.

Komponen	Spesifikasi
Virtualisasi	Ubuntu 18.04.3 LTS
Python	Python 3.8
Library	Tensorflow, Keras, sklearn, librosa, matplotlib, pandas
Komponen	Spesifikasi

Tabel 4.
Hasil Uji Coba Model CNN.

Model	*speech f1 score*	*non-speech f1 score*	Weighted Average
CNN (Sehgal)	0.8568	0.0382	0.6531
MobileNetV3-Small	0.8555	0.7224	0.8224
MobileNetV3-Large	**0.8652**	**0.7332**	**0.8324**

layer input, layer output, dan layer tersembunyi di antaranya seperti terlihat pada Gambar 5 [6].

3) Log-Mel Filterbank Energy Features

Untuk mentransformasi suara menjadi gambar, digunakan *log mel-filterbank energies. Log-mel energy spectrum* mewakili kekuatan jangka pendek dari data suara dalam *mel-frequency scale* selama beberapa durasi waktu. Spektrum energi *log-mel* terdiri dari *mel-frequency spectral coefficients* (MFSC). Koefisien ini mirip dengan *melfrequency cepstral coefficients* (MFCC). *mel scale of frequencies* menunjukkan skala persepsi frekuensi yang secara subjektif dinilai sama dalam jarak satu sama lain dalam hal sensasi pendengaran. Fungsi *B(f)* untuk menghitung frekuensi mel ke-*m* dari frekuensi atau *f* dalam Hz dapat dilihat pada Formula 1 dan inversnya $B^{-1}(m)$ memiliki formula yang dapat dilihat. Setelah menemukan koefisien MFSC sejumlah *N*, mereka digabungkan untuk membuat gambar $N \times B$, di mana *B* mewakili jumlah frame yang dipertimbangkan dalam spektrum. Gambar ini disebut log-mel energy spectrum [2]. Dalam pengerjaan digunakan library librosa Persamaan 1 sampai Persamaan 5 untuk mengubah data suara menjadi gambar dalam *log-mel energy spectrum.*

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

$$B^{-1}(m) = 700\left(10^{\frac{m}{2595}} - 1\right) \tag{2}$$

$$\hat{f}(n) = \frac{(K+1) * B^{-1}(\hat{m}(n))}{f_s}, n = 0 \dots N+1 \tag{3}$$

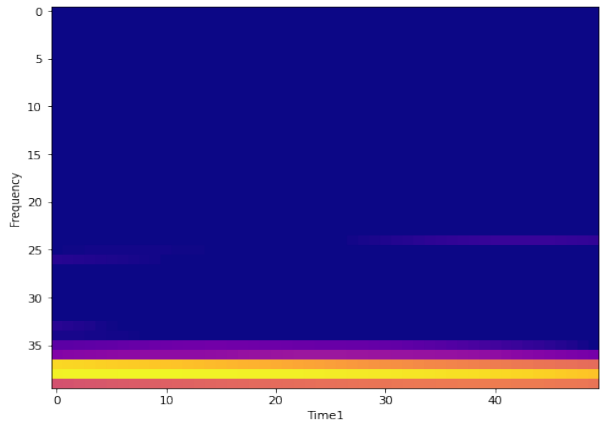
$$H_n(k) = \begin{cases} 0 & k < \hat{f}(n-1) \\ \dfrac{k - \hat{f}(n-1)}{\hat{f}(n) - \hat{f}(n-1)} & \hat{f}(n-1) < k \le \hat{f}(n) \\ \dfrac{\hat{f}(n+1) - k}{\hat{f}(n+1) - \hat{f}(n)} & \hat{f}(n) < k \le \hat{f}(n+1) \\ 0 & k > \hat{f}(n+1), \end{cases} \tag{4}$$

$k = 1 \dots K/2$
$n = 1 \dots N$

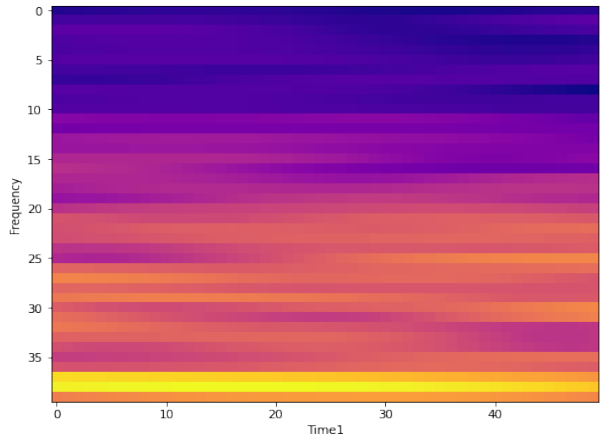
$$\text{MFSC}(n) = \log\left(\sum_{k=0}^{K} H_n(k) * |F(k)|^2\right), n = 1 \dots N \tag{5}$$

4) Signal to Noise Ratio

Signal-to-noise ratio adalah sebagai nilai numerik tunggal dalam desibel (dB). Yang dapat memiliki nilai positif atau negatif. SNR yang memiliki nilai positif menunjukkan bahwa



Gambar 9. Frame sebelum ditambah kebisingan.



Gambar 10. Frame setelah ditambah kebisingan.

Tabel 5.
Hasil Uji Kebisingan.

Kebisingan	Weighted Average
No Noise	0.8324
Café Loud	0.7948
Café Quiet	0.8125
Low Rumble	0.8097
Street Loud	0.8184
Street Quiet	0.8104

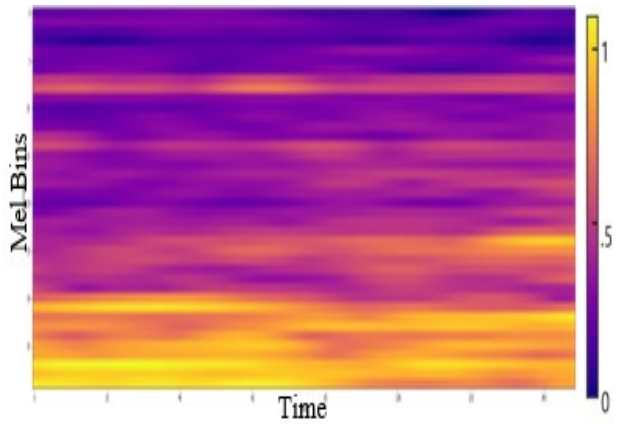
ucapan memiliki volume yang lebih besar di banding dengan tingkat kebisingan, atau data audio memiliki kebisingan rendah. Ketika nilai dari SNR adalah 0, maka ucapan memiliki volume yang sama yang sama dengan kebisingan. Dalam penelitian ini digunakan *decibel SNR*. Semua SNR data audio memiliki nilai positif. Hal ini menunjukkan bahwa semua data ucapan pada penelitian ini memiliki volume yang lebih besar dibandingkan dengan volume kebisingan.

$$SNR_{db} = signal_{db} - noise_{db} \quad (6)$$

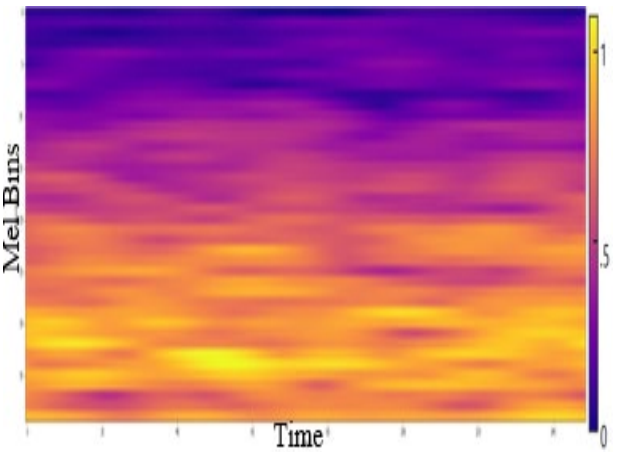
III. PERANCANGAN SISTEM

A. Perancangan Sistem Keseluruhan

Penelitian ini mengguanakan secara garis besar terdiri dari 4 langkah. Sebelum dataset dapat dipakai, tahap pemrosesan awal dataset dilakukan, lalu ekstraksi fitur *log-mel spectrogram* dari data audio untuk mempersiapkan input. Model CNN dibuat bedasarkan model-model yang ingin diuji. Terakhir, prediksi dan evaluasi dari model yang diuji seperti yang ditunjukkan bagan perancangan pada Gambar 6.



Gambar 11. Spektrogram *Café Loud.*



Gambar 12. Spektrogram *Street Loud.*

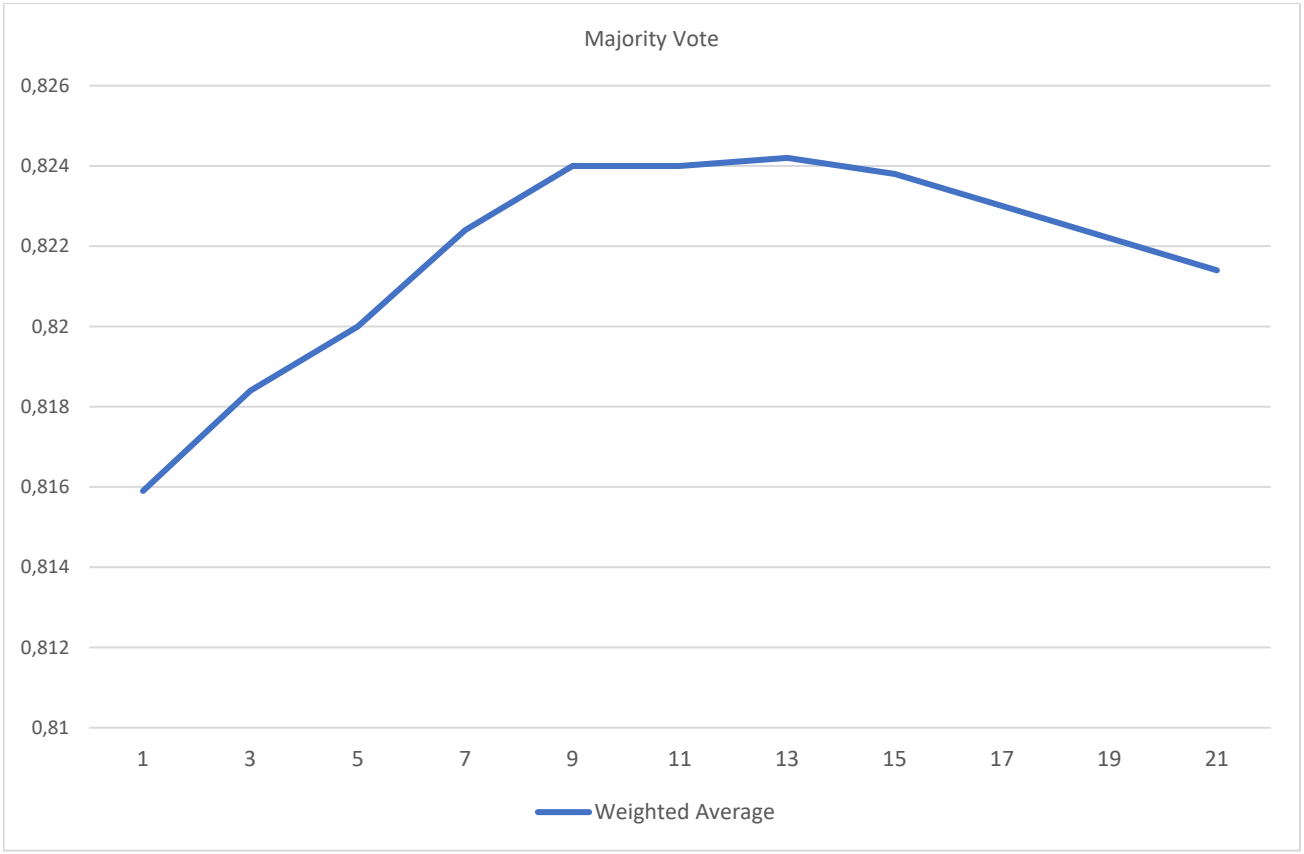
Tabel 6.
Hasil Uji *Hyperparameter.*

Optimizer	Learning Rate		
	0.1	0.01	0.001
SGD	0.8269	0.7858	0.6446
Adam	**0.8392**	0.8224	0.8276
RMSProp	0.6446	0.8361	0.8237

B. Pemrosesan Awal Dataset

Dataset yang telah diunduh dari *librispeech* berada dalam keadaan terpisah-pisah menjadi segmen 35 detik atau kurang. Dataset *librispeech* mematokkan jeda selama 0.5 detik di mana data audio tidak memiliki ucapan sebagai tempat pemisah segmen data audio [7]. Untuk menambah data audio sunyi, data yang terpisah ini digabung menjadi 1 file data audio panjang. Pada *data pipeline* dataset yang terdiri dari data audio dan data label diproses dijelaskan pada Gambar 7.

Tahap terakhir dari data audio yang sudah diubah menjadi *log-mel spectrogram* adalah membagi data spektrogram menjadi *frame* input model CNN yang memiliki 25ms dan *overlap* 50% atau *step* 12.5ms seperti yang ditunjukkan Gambar 8. Sehingga, hasil akhir dari pemrosesan data audio siap dipakai dalam model CNN. Hasil dataset ini terdiri dari *frame* yang berbentuk 1 *channel* berukuran 40 x 50. Hasil dataset ini tidak memiliki kendala saat menjadi input kepada dataset Model CNN Sehgal Namun, MobileNetV3 yang telah *pre-trained* dengan *imagenet* dataset membutuhkan input berbentuk 224 x 224 dengan 3 channel untuk RGB. Maka pemrosesan lanjutan untuk menyesuaikan diperlukan sebelum dataset dapat menjadi input dari MobileNetV3.



Gambar 13. Uji Coba Parameter Majority Vote.

C. *Pembuatan Model*

Model pertama adalah model CNN yang dapat dilihat pada Tabel 1. Model CNN ini diajukan pada penelitian sebelumnya oleh [2]. Model kedua yang diuji dalam penelitian ini adalah menggunakan MobileNetV3-Small dan MobileNetV3-Large.

D. *Prediksi dan Evaluasi*

Perancangan basis data merupakan satu tahap dalam pembuatan Prediksi hasil uji coba dilakukan dengan *input* berupa data uji yang dilakukan validasi data menggunakan model yang telah dihasilkan menggunakan data *training* sebelumnya dan menggunakan data *testing* yang berupa dataset data *log-mel spectrogram*. *Output* yang dihasilkan pada tahap ini berupa prediksi biner ada atau tidaknya ucapan pada *log-mel spectrogram*.

Setelah dilakukan prediksi hasil uji coba, maka dilakukan evaluasi model dan analisis dari hasil prediksi untuk mengetahui perbandingan kinerja dari model yang paling optimal untuk mendeteksi ucapan. Evaluasi performa model yang digunakan di sini adalah perhitungan nilai *Speech Hit Rate* dan *Noise Hit Rate*.

IV. UJI COBA DAN EVALUASI

Uji coba dilakukan dalam lingkungan *hardware* dan *software* seperti yang ditunjukkan pada Tabel 2 dan Tabel 3.

Masing-masing bedasarkan perancangan sistem ini akan dibagi menjadi 4 uji coba, yaitu:

1) *Uji Coba Model CNN*

Uji coba ini dijalankan dengan tujuan untuk menghitung performa model *CNN VAD* terhadap dataset *librispeech*. MobilNetV3-Large mendapatkan nilai 0.8324 yang terbaik diantara ketiga model. Pada penelitian ini MobileNetV3-Small digunakan selanjutnya karena performa yang tidak jauh beda dari MobileNetV3-Large dan supaya lebih banyak skenario yang dapat diuji coba. Dan hasil uji coba ditunjukkan pada Tabel 4.

2) *Uji Coba Kebisingan*

Uji coba pada skenario kedua bertujuan untuk mencari jenis kebisingan yang mengecoh model CNN. Hasil ekstraksi fitur dianalisa sebelum dan sesudah ditambahkan kebisingan seperti yang ditunjukkan Gambar 9 dan Gambar 10. Kebisingan yang terdapat pada frekuensi dapat dilihat adanya yang serupa dengan frekuensi ucapan, namun volume dari saat kebisingan lebih rendah dari pada volume data ucapan dengan hasil uji coba yang ditunjukkan pada Tabel 5.

Uji coba pada skenario kedua bertujuan untuk mencari jenis kebisingan yang memiliki kemampuan tertinggi untuk mengecoh model CNN deteksi ucapan. Selanjutnya, penambahan data kebisingan pada data audio dilakukan sehingga data audio memiliki tingkat kebisingan SNR 9db.

Performa terbaik kedua dihasilkan oleh penambahan kebisingan *street loud*. Performa terburuk terdapat pada data audio yang ditambah data kebisingan *low rumble* dan *café loud*. *Low Rumble, Café Loud* memiliki volume besar pada mel-bins 35 - 40, 20 atau kedua-duanya. Hal ini sama dengan *mel-bins* atau frekuensi data ucapan yang dapat dilihat pada Gambar 11 dan Gambar 12. Frekuensi dari data kebisingan yang sama dengan data ucapan akan meningkatkan kemampuan mengecoh dari data kebisingan.

3) *Uji Coba Hyperparameter*

Uji coba pada skenario ketiga bertujuan untuk mencari *hyperparameter* terbaik dengan model *MobilenetV3-Small*.

Hyperparameter yang dijadikan variabel untuk uji adalah pencobaan dengan *hyperparameter optimizer Adam* dan *learning rate* sebesar 0.1. dan hasil uji yang ditunjukkan pada Tabel 6.

1) Uji Coba Majority Vote

Uji coba skenario keempat bertujuan untuk mencari *smoothing* terbaik yang bisa digunakan setelah prediksi dari model ditemukan. Prediksi dari model menghasilkan nilai prediksi yang dapat berubah sangat cepat. Maka prediksi dari model dapat dioptimisasi lebih lanjut dengan *majority vote.* Hasil dari pengujian parameter *majority vote* mendapatkan parameter terbaik untuk *majority vote* adalah 13 seperti hasil uji coba yang ditunjukkan pada Gambar 13.

V. KESIMPULAN DAN SARAN

Kesimpulan yang di dapatkan berdasarkan Hasil Uji coba yang telah didapat adalah sebagai berikut : (1) Data audio dapat diubah menjadi data gambar menggunakan *log-mel spectrogram*; (2) Model MobileNetV3-Large menghasilkan performa terbaik untuk permasalahan deteksi ucapan.; (3) Metode evaluasi yang digunakan dalam deteksi ucapan Model CNN menggunakan *speech f1 score*, *non-speech f1 score*, dan *weighted average*; (4) Model MobileNetV3 memiliki performa terbaik saat menggunakan hyperparameter optimizer Adam dan learning rate 0.1; (6) Metode smoothing yang digunakan pada penelitian ini adalah majority vote 13 *optimizer* dan *learning rate*. Hasil dari uji coba pada skenario

Saran yang di dapat dari uji coba dan evaluasi adalah sebagai berikut : (1) Perbanyak data audio ucapan dengan variasi bahasa, intonasi atau emosi; (2) Eksraksi fitur log mel spectrogram dapat diuji beberapa parameter berbeda sehingga model CNN mendapatkan konteks lebih banyak; (3) Menghilangkan *outlier* dari data audio dan data kebisingan yang bernilai ekstrim; (4) Perbanyak Model CNN yang diuji, dan Perbanyak data kebisingan agar model memiliki performa baik pada kondisi beragam

DAFTAR PUSTAKA

[1] J. Sohn, N. S. Kim, and W. Sung, “A statistical model-based voice activity detection,” *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, 1999.

[2] A. Sehgal and N. Kehtarnavaz, “A convolutional neural network smartphone app for real-time voice activity detection,” *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[3] C. Kim *et al.*, “Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” *INTERSPEECH*, 2017.

[4] M. Huzaifah, “Comparison of time-frequency representations for environmental sound classification using convolutional neural networks,” *arXiv Prepr. arXiv1706.07156*, 2017.

[5] A. Howard *et al.*, “Searching for Mobilenetv3,” in *Proceedings Of The IEEE/CVF International Conference On Computer Vision*, 2019, pp. 1314–1324.

[6] S. Saha, “A comprehensive guide to convolutional neural networks—the ELI5 way,” *Towar. data Sci.*, vol. 15, p. 15, 2018.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An Asr Corpus Based On Public Domain Audio Books,” in *2015 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*, 2015, pp. 5206–5210.