Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.20241127866
Vol. 11, No. 2, April 2024, hlm. 369-376 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

SISTEM IDENTIFIKASI PEMBICARA BERBAHASA INDONESIA MENGGUNAKAN X-VECTOR EMBEDDING

Alim Misbullah[*1], Muhammad Saifullah Sani[2], Husaini[3], Laina Farsiah[4], Zahnur[5], Kikye Martiwi Sukiakhy[6]

[1,2,3,4,5,6] Universitas Syiah Kuala, Banda Aceh
Email : [1]misbullah@usk.ac.id, [2]saiful19@mhs.usk.ac.id, [3]husaini.muhammad@usk.ac.id, [4]lainafarsiah@usk.ac.id, [5]zahnur@usk.ac.id, [6]kikye.martiwi.sukiakhy@usk.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 14 Oktober 2023, diterima untuk diterbitkan: 04 April 2024)

Abstrak

Penyemat pembicara adalah vektor yang terbukti efektif dalam merepresentasikan karakteristik pembicara sehingga menghasilkan akurasi yang tinggi dalam ranah pengenalan pembicara. Penelitian ini berfokus pada penerapan x-vectors sebagai penyemat pembicara pada sistem identifikasi pembicara berbahasa Indonesia yang menggunakan model speaker identification. Model dibangun dengan menggunakan dataset VoxCeleb sebagai data latih dan dataset INF19 sebagai data uji yang dikumpulkan dari suara mahasiswa Jurusan Informatika Universitas Syiah Kuala angkatan 2019. Untuk membangun model, fitur-fitur diekstrak dengan menggunakan *Mel-Frequency Cepstral Coeffients* (MFCC), dihitung *Voice Activity Detection* (VAD), dilakukan augmentasi dan normalisasi fitur menggunakan *Cepstral Mean* and *Variance Normalization* (CMVN) serta dilakukan filtering. Sedangkan proses pengujian model hanya membutuhkan fitur-fitur yang diekstrak dengan menggunakan MFCC dan dihitung VAD saja. Terdapat 4 (empat) model yang dibangun dengan cara mengombinasikan dua jenis konfigurasi MFCC dan dua jenis arsitektur *Deep Neural Network* (DNN) yang memanfaatkan *Time Delay Neural Network* (TDNN). Model terbaik dipilih berdasarkan akurasi tertinggi yang dihitung menggunakan metrik *Equal Error Rate* (EER) dan durasi ekstraksi x-vectors tersingkat dari keempat model. Nilai EER dari model yang terbaik untuk dataset VoxCeleb1 bagian test sebesar 3,51%, inf19_test_td sebesar 1,3%, dan inf19_test_tid sebesar 1,4%. Durasi ekstraksi x-vectors menggunakan model terbaik untuk data train berdurasi 6 jam 42 menit 39 detik, VoxCeleb1 bagian test berdurasi 2 menit 24 detik, inf19_enroll berdurasi 18 detik, inf19_test_td berdurasi 25 detik, dan inf19_test_tid berdurasi 9 detik. Arsitektur DNN kedua dan konfigurasi MFCC kedua yang telah dirancang menghasilkan model yang lebih kecil, akurasi yang lebih baik terutama untuk dataset pembicara berbahasa Indonesia, dan durasi ekstraksi x-vectors yang lebih singkat.

Kata kunci: *identifikasi pembicara, time delay neural network, x-vectors, mel frequency cepstrum coefficient, equal error rate*

SPEAKER IDENTIFICATION FOR INDONESIAN LANGUAGE USING X-VECTOR EMBEDDING

Abstract

The speaker embedding is a vector that has been proven effective in representing speaker characteristics, resulting in high accuracy in the domain of speaker recognition. This research focuses on the application of x-vectors as speaker embeddings in the Indonesian language speaker identification system using a speaker identification model. The model is built using the VoxCeleb dataset as training data and the INF19 dataset as testing data, collected from the voices of students of Informatics Department, Universitas Syiah Kuala from the 2019 batch. To build the model, features are extracted using Mel-Frequency Cepstral Coefficients (MFCC), Voice Activity Detection (VAD) is applied, augmentation and normalization of features are performed using Cepstral Mean and Variance Normalization (CMVN), and filtering is applied. On the other hand, the model testing process only requires features extracted using MFCC and computed VAD. There are 4 (four) models are constructed by combining two configurations of MFCC and two types of Deep Neural Network (DNN) architectures that utilize the Time Delay Neural Network (TDNN). The best model is selected based on the highest accuracy calculated using the Equal Error Rate (EER) metric and the shortest duration of x-vector extraction from the four models. The EER values for the best model on the VoxCeleb1 test dataset are 3.51%, 1.3% for inf19_test_td, and 1.4% for inf19_test_tid. The x-vector extraction duration using the best model for the training dataset is 6 hours 42 minutes 39 seconds, 2 minutes 24 seconds for VoxCeleb1 test part, 18 seconds for inf19_enroll, 25 seconds for

inf19_test_td, and 9 seconds for inf19_test_tid. The second DNN architecture and the second MFCC configuration designed result in a smaller model, better accuracy, especially for Indonesian language speaker datasets, and shorter x-vector extraction duration.

Keywords: *speaker identification, time delay neural network, x-vectors, mel frequency cepstrum coefficient, equal error rate*

1. PENDAHULUAN

Speaker recognition telah digunakan pada banyak alat-alat canggih, yaitu mobil pintar, laptop, ponsel cerdas, dan lainnya. Contoh penggunaannya, seperti pada alat yang digunakan pada bidang forensik untuk menyelidiki tersangka salah atau tidak, autentikasi berbasis suara pada piranti cerdas, seperti telepon genggam, kendaraan, dan laptop, rekaman rapat dan panggilan telepon, keamanan transaksi antar bank, pembayaran jarak jauh, dan sebagai *frontend* pada *automatic speech recognition* agar performa transkripsi yang dihasilkan menjadi lebih baik pada percakapan banyak pembicara (Bai & Zhang, 2021). *Speech recognition* atau pengenalan ucapan adalah proses ekstraksi dan menentukan informasi linguistik yang didapat dari gelombang suara dengan menggunakan komputer atau sirkuit elektronik. Singkatnya, pengenalan ucapan bertujuan hanya untuk mengenali informasi linguistik. Sedangkan pengenalan pembicara adalah proses mengenali siapa yang berbicara dengan menggunakan informasi khusus pembicara yang terdapat dalam gelombang suara untuk memverifikasi identitas orang yang terdaftar di sistem. Tujuan dari pengenalan pembicara adalah untuk mendapatkan informasi individu yang menunjukkan *who is speaking* atau siapa yang sedang berbicara (Furui, 2018).

Tahapan atau fase dalam sistem pengenalan pembicara terdiri dari dua fase, yaitu fase *enrollment* (pendaftaran) dan fase *testing*. Pada fase pendaftaran, pembicara mendaftarkan suaranya agar dikenali oleh sistem. Pada fase *testing*, pembicara yang telah terdaftar memasukkan suaranya ke dalam sistem dan model akan berusaha menentukan pemilik suara tersebut berdasarkan data pembicara yang telah didaftarkan dan suara yang dimasukkan. Jika kata-kata yang digunakan pada kedua fase harus sama, maka dikategorikan sebagai *text-dependent* dan jika tidak, maka dikategorikan sebagai *text-independent* (Xiang et al., 2019). Berdasarkan cara kerjanya, pengenalan pembicara dibagi menjadi *speaker verification* dan *speaker identification* (Furui, 2018). *Speaker verification* atau verifikasi pembicara adalah prosedur memverifikasi identitas pembicara yang diklaim berdasarkan sinyal ucapan dari pembicara (Cao et al., 2018). Secara lebih rinci, verifikasi pembicara adalah tugas memverifikasi apakah ucapan yang ingin diuji dan ucapan yang terdaftar berasal dari pembicara yang sama. Hal ini dilakukan dengan cara membandingkan nilai kemiripan dari kedua ucapan dengan suatu batas yang telah didefinisikan (Bai & Zhang, 2021). *Speaker verification* bertujuan untuk menentukan apakah suara pembicara yang sedang berbicara cocok dengan pembicara yang telah didaftarkan di sistem (Ding et al., 2020). Sedangkan *speaker identification* atau identifikasi pembicara adalah tugas menentukan identitas pembicara dari sekumpulan pembicara berdasarkan ucapan yang diuji (Bai & Zhang, 2021). Tujuan identifikasi pembicara adalah untuk mengidentifikasi pembicara suatu ucapan dari kumpulan pembicara yang telah dikenal oleh sistem (Ding et al., 2020).

Feature extraction adalah proses untuk menentukan nilai atau vektor yang dapat digunakan sebagai pembeda antar objek atau individu. Dalam ranah suara, biasanya algoritma ekstraksi fitur yang digunakan adalah *(*MFCC) yang menghitung koefisien cepstral dengan mempertimbangkan persepsi sistem pendengaran manusia terhadap frekuensi suara, yaitu filter sinyal yang bersifat linier terhadap frekuensi di bawah 1 kHz dan bersifat logaritmik terhadap frekuensi di atas 1 kHz. Metode ini lebih unggul dibandingkan metode lainnya karena dapat menangkap karakteristik atau informasi penting pembicara dari sinyal suara dan menghasilkan data seminimal mungkin tanpa menghilangkan informasi penting yang terkandung di dalam sinyal suara (Nursholihatun, 2020).

Deep Neural Networks (DNN) adalah pengembangan lebih lanjut dari *Artificial Neural Network* (ANN). Secara umum, *neural network* terdiri dari beberapa layer atau lapisan, yaitu input layer, hidden layer, dan output layer, yang masing-masing lapisan memiliki sejumlah node. Kumpulan node yang ada pada input layer adalah vektor representasi dari data input. Kumpulan node yang ada pada hidden layer berguna sebagai penentu informasi dari lapisan input untuk diteruskan ke lapisan berikutnya. Kumpulan node yang ada pada output layer berperan sebagai label atau kelas yang akan diprediksi (Misbullah et al., 2020). Perbedaan DNN dengan ANN adalah penambahan hidden layer yang lebih banyak dan bervariasi. Oleh karena itu, DNN sangat cocok digunakan dalam mempelajari sesuatu yang lebih rumit dan kompleks daripada ANN (Cao et al., 2018). Salah satu arsitektur DNN adalah *Time Delay Neural Networks* (TDNN). Menurut penelitian (Peddinti et al., 2015), TDNN membutuhkan waktu pemrosesan yang lebih singkat daripada arsitektur *Recurrent Neural Network* (RNN), contohnya *Long Short-Term Memory* (LSTM), karena TDNN tidak menggunakan hubungan antara panjang konteks

input dan jumlah langkah berurutan selama pelatihan. Unit dasar TDNN berupa modifikasi dari unit dasar DNN dengan memperkenalkan delay yang bermaksud agar input sekarang akan dikalikan dengan beberapa bobot yang berasal dari input-input sebelumnya. Dengan demikian, TDNN mampu menghubungkan dan membandingkan input sekarang dengan riwayat input-input kejadian sebelumnya (Fan et al., 2021). Oleh karena itu, penelitian ini menggunakan arsitektur DNN dengan memanfaatkan TDNN sehingga nantinya akan diekstrak *x-vectors*, yaitu vektor yang merepresentasikan pembicara.

Data speech pada dunia nyata tidak hanya berisi seseorang yang sedang berbicara. Oleh karena itu, perlu penyemat pembicara yang berguna untuk menangkap karakteristik pembicara (Xie et al., 2019). *X-vectors* adalah embedding atau penyemat yang diekstrak dari DNN dengan menggunakan arsitektur TDNN yang sangat efektif untuk pengenalan pembicara dan speaker diarization (Xie et al., 2019). *Speaker embedding* atau penyemat pembicara tingkat pembicara yang bernama x-vectors ini memiliki performa yang lebih baik daripada i-vector yang telah menjadi algoritma paling mutakhir untuk waktu yang cukup lama dalam ranah pengenalan pembicara (Xiang et al., 2019). Model yang berkenaan dengan biometrik biasanya dievaluasi menggunakan metrik *Equal Error Rate* (EER) (Chung et al., 2018).

Kaldi adalah salah satu toolkit yang digunakan untuk penelitian pengenalan suara dan pengenalan pembicara yang bersifat open-source yang dibangun menggunakan bahasa pemrograman C++ dan di bawah lisensi Apache License v2.0. Kaldi sudah menyediakan banyak contoh proyek yang disebut recipe dan model yang disediakan berbasis *Finite-State Transducers* (FST). Sampai saat ini, Kaldi belum menyediakan pre-trained model yang berfokus untuk bahasa Indonesia. Tujuan utama Kaldi adalah kode yang modern dan fleksibel sehingga mudah dipahami, dimodifikasi, dan diperluas (Povey et al., 2011).

2. METODE PENELITIAN

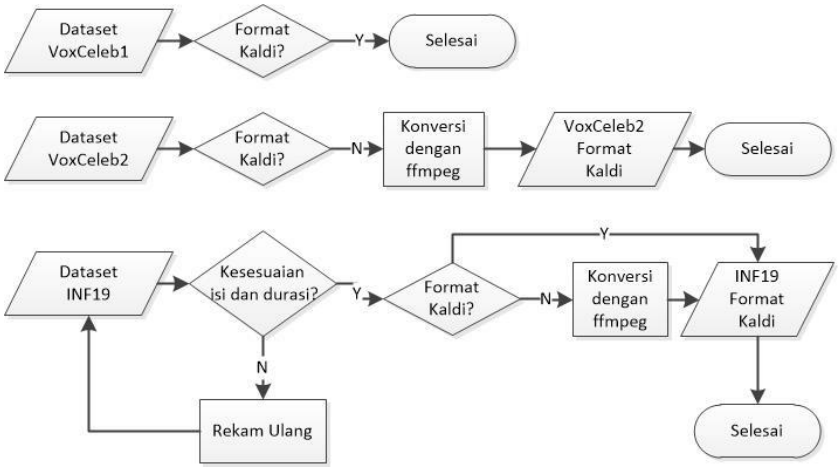
Penelitian ini menggunakan toolkit Kaldi dengan mengikuti recipe Voxceleb v2. Diagram alir dari penelitian ini dapat dilihat pada Gambar 1. Dimulai dari pengumpulan dataset yang berupa berkas-berkas audio, pre-processing data audio, persiapan berkas dalam format Kaldi, ekstraksi fitur, pembangunan model, pengujian model, dan analisis hasil pengujian model. Dataset utama yang digunakan adalah Voxceleb1, VoxCeleb2, dan INF19. Sedangkan dataset RIRs Noise dan MUSAN hanya digunakan untuk melakukan augmentasi. Setelah dataset utama dikumpulkan, dilakukan pre-processing berkas audio agar dapat diterima oleh Kaldi. Lalu, disiapkan berkas-berkas dalam format Kaldi agar dapat diproses. Selanjutnya, digunakan dua konfigurasi MFCC untuk ekstraksi fitur dan dihitung VAD. Khusus dataset untuk pembangunan model, dilakukan augmentasi menggunakan dataset RIRs Noise dan MUSAN, normalisasi fitur menggunakan CMVN, dan filtering atau penyaringan. Setelah fitur-fitur sudah tersedia, dibangun empat model dengan menggunakan kombinasi dari dua konfigurasi MFCC dan dua arsitektur DNN yang memanfaatkan TDNN. Setiap model yang telah dibangun diuji menggunakan data pengujian dengan memanfaatkan x-vectors yang diekstrak menggunakan masing-masing model dan dianalisis hasil pengujian model untuk mencari model terbaik berdasarkan akurasi tertinggi dan durasi ekstraksi x-vectors tersingkat.

2.1 Pengumpulan Dataset

Dataset yang digunakan berupa berkas audio dari dataset VoxCeleb1, VoxCeleb2, RIRs Noise, MUSAN, dan INF19. Dataset utama yang digunakan adalah Voxceleb1, VoxCeleb2, dan INF19. Sedangkan dataset RIRs Noise dan MUSAN hanya digunakan untuk melakukan augmentasi. Dataset VoxCeleb1, VoxCeleb2, RIRs Noise, dan MUSAN tersedia secara publik , sedangkan dataset INF19 dibangun dengan menggunakan rekaman suara mahasiswa dan mahasiswi Informatika Universitas Syiah Kuala angkatan 2019 yang mengucapkan 6 (enam) kalimat yang telah dirancang, yaitu:

1. Saya adalah salah satu mahasiswa Informatika angkatan 2019.
2. Informatika adalah salah satu jurusan di fakultas MIPA.
3. MIPA adalah salah satu fakultas di Universitas Syiah Kuala.
4. Universitas Syiah Kuala terletak di Darussalam, kota Banda Aceh.
5. Subhanallah walhamdulillah wala ilaha illallah wallahu akbar.

Gambar 1. Alur Tahapan Penelitian



Gambar 2. Langkah-Langkah Dalam Pre-processing Data

Untuk kalimat pertama hingga kalimat kelima diulang sebanyak 5 (lima) kali dalam rekaman yang berbeda dan disimpan dalam berkas yang berbeda pula. Kalimat keenam berupa kalimat dan bahasa yang bebas selain kelima kalimat di atas dan diulang sebanyak 5 (lima) kali atau mengucapkan 5 (lima) kalimat yang berbeda-beda yang masing-masing diucapkan sekali. Peraturan kalimat keenam ini bertujuan untuk mendapatkan rekaman suara pembicara dalam bentuk kalimat, bahasa, gaya, dan intonasi yang biasa digunakan sehari-hari. Untuk kalimat keenam, diperoleh 68 pembicara menggunakan bahasa Indonesia, 3 pembicara menggunakan bahasa Arab, dan seorang pembicara menggunakan bahasa Aceh. Hanya seorang pembicara yang mengucapkan lima kalimat berbeda pada rekaman untuk kalimat keenam. Jumlah berkas audio dataset INF19 yang terkumpul adalah sebanyak 2.160 berkas audio yang berasal dari 72 pembicara dengan durasi total sekitar 3 jam 30 menit.

Dataset VoxCeleb1 dan VoxCeleb2 (Cao et al., 2018; Ding et al., 2020; Xiang et al., 2019) terbagi menjadi bagian dev dan test. Dataset INF19 terbagi menjadi inf19_enroll, inf19_test_td, dan inf19_test_tid. Untuk pembagian dataset, VoxCeleb1 bagian dev, VoxCeleb2 bagian dev, dan VoxCeleb2 bagian test disebut sebagai data train. Pembangunan model menggunakan data train yang diaugmentasi menggunakan dataset RIRs Noise dan MUSAN serta dilakukan penyaringan. Pengujian model terbagi menjadi tahap pendaftaran dan testing. Dataset yang digunakan pada tahap pendaftaran adalah dataset data train dan inf19_enroll yang disebut sebagai dataset data enroll. Sedangkan dataset yang digunakan pada tahap testing adalah VoxCeleb1 bagian test, inf19_test_td, dan inf19_test_tid yang disebut sebagai data testing. Gabungan dari dataset data train, inf19_enroll, dan data testing disebut sebagai data test. Jumlah berkas audio dan jumlah pembicara dari dataset utama dapat dilihat pada Tabel 1.

Tabel 1. Distribusi Jumlah Audio dan Pembicara

Dataset	Jumlah Berkas Audio	Jumlah Pembicara
VoxCeleb1 *dev*	148.642	1.211
VoxCeleb1 *test*	4.874	40
VoxCeleb2 *dev*	1.092.009	5.994
VoxCeleb2 *test*	36.237	118
inf19_enroll	600	60
inf19_test_td	1.080	72
inf19_test_tid	360	72

2.2 Pre-processing Data Audio

Semua berkas audio dalam dataset utama harus sesuai dengan format audio yang diterima oleh Kaldi, yaitu berekstensi WAV dengan channel yang digunakan adalah mono channel, sample rate sebesar 16 kHz, dan sample size sebesar 16 bit. Dataset VoxCeleb1 sudah dalam format yang sesuai dengan yang dibutuhkan Kaldi sehingga dapat langsung dibuatkan berkas-berkas yang dibutuhkan dalam format Kaldi dengan menggunakan script di Kaldi. Dataset VoxCeleb2 perlu dilakukan pre-processing karena berkas audio tidak dalam format yang sesuai dengan yang dibutuhkan Kaldi. Pre-processing dataset VoxCeleb2 dilakukan dengan cara mengonversi setiap berkas audio ke dalam format yang diterima Kaldi dengan menggunakan script di Kaldi yang memanfaatkan aplikasi *ffmpeg*. Untuk dataset INF19, diperiksa kalimat yang diucap, durasi audio, nama folder speaker id, nama folder utterance id, nama berkas audio, dan format audio yang dapat diterima oleh Kaldi. Ketidaksesuaian kalimat yang diucap dan durasi audio diatasi dengan cara meminta pemilik rekaman untuk merekam ulang. Ketidaksesuaian ekstensi berkas audio, channel, sample rate, dan sample size diatasi dengan cara mengonversi berkas audio tersebut menggunakan aplikasi *ffmpeg* agar sesuai dengan format yang dibutuhkan Kaldi.Setelah selesai dilakukan pre-processing dataset INF19, dataset tersebut dipisah ke dalam tiga folder, yaitu “inf19_enroll”, “inf19_test_td”, dan “inf19_test_tid”. Pemisahan ini digunakan untuk tahap analisis hasil pengujian model menggunakan dataset INF19. Pembagian target speaker dan impostor pada dataset INF19 adalah 60

pembicara pertama sebagai target speaker dan sisanya, yaitu 12 pembicara dijadikan sebagai impostor. Untuk pembagian target speaker dan impostor pada pengujian menggunakan dataset VoxCeleb1 bagian test mengikuti pengaturan yang telah disediakan di recipe Voxceleb2, yaitu data train digunakan sebagai target speaker dan VoxCeleb1 bagian test digunakan sebagai impostor.

2.3 Ekstraksi Fitur

Setelah berkas-berkas yang dibutuhkan Kaldi telah tersedia, dilakukan augmentasi data train dengan menggunakan dataset RIRs Noise dan MUSAN yang hasilnya disebut sebagai data train aug. Agumentasi ini bertujuan untuk menambah jumlah dan variasi dari kondisi pada audio yang digunakan untuk membangun model agar model yang dibangun menjadi lebih baik. Data train berjumlah 1.276.888 berkas audio dan data train aug berjumlah 5.107.552 berkas audio.

Kemudian, diekstrak fitur-fitur dari dataset data test dan sejuta berkas audio dari data train aug dengan menggunakan dua konfigurasi MFCC yang berbeda yang dapat dilihat pada Tabel 2. Selanjutnya, VAD digunakan pada dataset data test untuk mendapatkan bagian yang mengandung ucapan saja. Sejuta data train aug tidak digunakan VAD karena hasil augmentasi akan menghasilkan gangguan atau bagian tidak berisi ucapan yang bertujuan agar kondisi suara pada data audio untuk pembangunan model menjadi lebih bervariasi. Konfigurasi VAD yang digunakan dapat dilihat pada Tabel 3. Hasil ekstraksi fitur data train dari data test dan sejuta data train aug digabung menjadi data train combined yang berjumlah 2.026.888 berkas audio. Sebagian berkas audio yang hilang disebabkan karena gagal melakukan ekstraksi MFCC dari berkas audio tersebut.

Tabel 2. Konfigurasi MFCC

Parameter	MFCC Pertama	MFCC Kedua
sample-frequency	16000 Hz	8000 Hz
frame-length	25 ms	-
low-freq	20 Hz	20 Hz
high-feq	7600 Hz	3700 Hz
num-mel-bins	30	-
num-ceps	30	23
allow-downsample	-	true

Selanjutnya, dilakukan normalisasi fitur data train combined menggunakan CMVN dan dilakukan penyaringan dengan menghapus berkas audio dari berkas utt2spk, spk2utt, dan wav.scp yang durasinya kurang dari empat detik setelah dihilangkan bagian yang hening dan pembicara yang memiliki kurang dari delapan berkas audio. Tujuannya adalah fitur-fitur yang digunakan untuk membangun model berupa fitur-fitur yang bagus. Hasil dari normalisasi fitur dan penyaringan terhadap data train combined disebut sebagai data train combined no sil yang berjumlah 1.894.294 berkas audio dan data ini yang digunakan untuk membangun model.

Tabel 3. Konfigurasi VAD

Parameter	Nilai
vad-energy-threshold	5.5
vad-energy-mean-scale	0.5
vad-proportion-threshold	0.12
vad-frames-context	2

2.4 Membangun Model

Dibangun empat model speaker identification dengan menggunakan dua arsitektur DNN untuk setiap konfigurasi MFCC. Dua konfigurasi MFCC yang digunakan dapat dilihat pada Tabel 2 dan dua arsitektur DNN yang digunakan dapat dilihat pada Tabel 4. Hyperparameter untuk membangun setiap model menggunakan aturan bawaan dari recipe Voxceleb v2 di Kaldi, yaitu learning rate sebesar 0,0001, minibatch size sebesar 64, dan epoch sebesar 3. Fungsi aktivasi yang digunakan pada setiap lapisan tersembunyi adalah fungsi rectified linear unit (ReLU) dan fungsi aktivasi yang digunakan pada lapisan keluaran adalah softmax.

Tabel 4. Arsitektur DNN

Lapisan	Jumlah *Node*	
	Arsitektur DNN Pertama	Arsitektur DNN Kedua
frame1	512	256
frame2	512	256
frame3	512	256
frame4	512	256
frame5	1.500	750
stats pooling	3.000	1.500
segment6	512	256
segment7	512	256

Pada Tabel 4, arsitektur DNN pertama adalah arsitektur DNN bawaan yang telah disediakan di recipe Voxceleb v2 (Chung et al., 2018) dan arsitektur DNN kedua adalah arsitektur DNN bawaan recipe Voxceleb v2 yang dimodifikasi dengan cara mengurangi jumlah node pada setiap lapisan sebanyak setengahnya. Empat model yang dibangun diberi nama Model-I, Model-II, Model-III, dan Model-IV. Keterangan kombinasi konfigurasi MFCC dan arsitektur DNN setiap model dapat dilihat pada Tabel 5.

2.5 Pengujian Model

Pengujian model dilakukan dengan cara menghitung akurasi dan durasi esktraksi x-vectors menggunakan model speaker identification yang dibangun. X-vectors diekstrak dari lapisan segment6. Dataset yang digunakan untuk perhitungan akurasi model menggunakan data testing dan perhitungan durasi ekstraksi x-vectors menggunakan data test. Perhitungan akurasi model menggunakan metrik EER dengan memanfaatkan script yang telah tersedia di Kaldi. Semakin kecil nilai EER dan semakin singkat durasi ekstraksi x-vectors, semakin baik performa model tersebut. Perhitungan durasi ekstraksi x-vectors memanfaatkan fungsi date di dalam pustaka bahasa pemrograman Shell Script.

Tabel 5. Kombinasi Konfigurasi MFCC dan Arsitektur DNN

Nama Model	Konfigurasi MFCC	Arsitektur DNN
Model-I	Pertama	Pertama
Model-II	Pertama	Kedua
Model-III	Kedua	Pertama
Model-IV	Kedua	Kedua

Pengujian model terbagi menjadi dua tahap, yaitu enrollment (pendaftaran) dan testing. Jika kata-kata yang digunakan oleh pembicara pada dua tahap ini harus sama, maka dikategorikan text-dependent dan jika tidak, maka dikategorikan text-independent. Dataset inf19_test_td digunakan untuk pengujian text-dependent, sedangkan dataset VoxCeleb1 bagian test dan inf19_test_tid digunakan untuk pengujian text-independent.

Pada tahap pendaftaran, diekstrak x-vectors dari pembicara yang ingin didaftarkan ke dalam sistem (target speaker), yaitu dataset data train untuk pengujian menggunakan dataset VoxCeleb1 bagian test dan inf19_enroll untuk pengujian menggunakan dataset inf19_test_td dan inf19_test_tid. Pada tahap testing, x-vectors dari data enroll yang sudah diekstrak dilakukan centering untuk mencari nilai-nilai rata-rata dari x-vectors yang telah diekstrak untuk merepresentasikan masing-masing pembicara. Disini, LDA digunakan untuk mengecilkan dimensi x-vectors dan model PLDA dilatih untuk menebak pemilik suara pada dataset VoxCeleb1 bagian test, inf19_test_td, dan inf19_test_tid agar dapat digunakan untuk perhitungan skor pada saat menghitung akurasi model DNN.

Selanjutnya, diekstrak x-vectors dari berkas-berkas audio para target speaker sebagai pengecoh (impostor) untuk dilakukan perhitungan skor menggunakan model PLDA yang telah dilatih. X-vectors yang diekstrak tersebut dimasukkan sebagai input ke dalam model PLDA yang telah dilatih untuk ditebak siapa pemilik suaranya. Model PLDA akan menebak target speaker pada data testing dengan cara mengeluarkan probabilitas yang tinggi untuk data target speaker dan probabilitas rendah untuk data impostor yang bermakna model dapat mengenali pengguna yang terdaftar di dalam sistem dan tidak terkecoh dengan data pengecoh. Terakhir, dihitung nilai EER dengan memanfaatkan hasil tebakan model PLDA yang telah dilatih untuk menghitung skor berdasarkan berkas-berkas verifikasi.

3. PENELITIAN TERKAIT

Penelitian mengenai pengenalan pembicara telah banyak dilakukan dengan menggunakan berbagai metode. Penelitian yang dilakukan oleh Chung pada tahun 2018 adalah membangun dataset VoxCeleb2 dengan cara mengekstrak suara seorang pembicara pada video-video di YouTube dengan mengimplementasikan ilmu computer vision untuk memilih pembicara pada video (Chung et al., 2018). Dataset tersebut berisi lebih dari sejuta utterance dari sekitar 6.000 selebriti dari berbagai usia, etnis, aksen, dan kondisi audio. Dataset yang dibangun digunakan untuk membandingkan model i-vectors dengan menggunakan PLDA sebagai backend, VGG-M, ResNet-34, dan ResNet-50 tanpa tahap pre-processing dan mengujinya menggunakan dataset VoxCeleb1. Hasil dari penelitian tersebut adalah model ResNet-50 memiliki performa terbaik, yaitu minimal cost dengan Ptar 0,01 bernilai 0,429 dan EER sebesar 3,95%.

Penelitian yang dilakukan oleh Snyder pada tahun 2018 mengenai speaker embedding yang diekstrak dari DNN yang dapat merepresentasikan karakteristik pembicara dengan sangat baik dan mampu mengalahkan metode ¬state-of-the-art dalam waktu yang cukup lama, yaitu i-vectors (Snyder et al., 2018). Pada tahap pembangunan model DNN, digunakan arsitektur TDNN untuk membangun model, metode MFCC untuk ekstraksi fitur, speech activity detection untuk menyaring frame yang tidak berisi ucapan dan menggunakan dataset SWBD, NIST SRE dari tahun 2004 hingga tahun 2010, VoxCeleb1, dan data augmentasi dari dataset RIRs Noise dan MUSAN untuk menambahkan data audio yang berisi gangguan ucapan orang lain, musik, kebisingan, dan gema. Model i-vectors dibandingkan dengan x-vectors dengan menggunakan probabilistic linear discriminant analysis (PLDA) untuk menghitung skor yang mana dimensinya terlebih dahulu dikurangi menggunakan linear discriminant analysis (LDA) dan menggunakan dataset SITW Core dan NIST SRE 2016 (SRE16). Hasil dari penelitian tersebut adalah x-vectors memiliki performa yang lebih baik daripada i-vectors dengan EER untuk data uji SITW Core bernilai 4,16% dan untuk data uji SRE16 bernilai 5,71%. Penelitian ini membuktikan bahwa x-vectors adalah speaker embedding yang telah menjadi ¬state-of-the-art untuk sekarang ini.

Pada penelitian yang dilakukan oleh Xie pada tahun 2019, mereka menggabungkan arsitektur thin-ResNet dengan layer NetVLAD atau GhostVLAD dan membandingkannya dengan model yang terkenal, yaitu i-vectors dengan PLDA sebagai backend, VGG-M, x-vectors, dan ResNet-20 (Xie et al., 2019). Model-model tersebut dibangun dengan menggunakan data bagian dev dari dataset VoxCeleb2 dengan menggunakan ResNet-34 untuk ekstraksi fitur dan dilakukan evaluasi model menggunakan dataset VoxCeleb1 bagian test. Disimpulkan bahwa thin ResNet-34 adalah model terbaik dengan EER sebesar 3,22% untuk data uji VoxCeleb1 test set, 3,13% untuk data uji VoxCeleb1-E, dan 5,06% untuk data uji VoxCeleb1-H. Selanjutnya, model terbaik ini diuji dengan data yang sudah dimodifikasi yang memiliki durasi dari dua detik hingga enam detik. Disimpulkan bahwa semakin panjang durasi pengucapan (minimal empat detik), maka performa model akan menjadi lebih baik.

Pada penelitian yang dilakukan oleh Fan pada tahun 2021 membahas mengenai perbandingan

performa DNN, bidirectional long short term memory (BLSTM), dan TDNN (Fan et al., 2021). Dataset yang digunakan untuk training, validation, dan test berasal dari dataset TIMIT yang sudah mengandung noise. Khusus untuk data test, ditambahkan noise baru dari dataset NISEX-92. Hasil yang didapat dari penelitian tersebut adalah TDNN memiliki performa yang lebih baik daripada DNN dan BLSTM untuk semua kondisi pengujian. Hasil dari TDNN memiliki kemampuan merekam konteks dalam waktu yang panjang dan waktu yang dibutuhkan untuk melakukan inference dapat dibandingkan dengan DNN standar. Penelitian itu juga membahas bahwa metode RNN memiliki komputasi yang lebih kompleks dan membutuhkan waktu yang lebih lama untuk training dan inference dibandingkan feed-forward DNN. Berdasarkan penelitian ini, disimpulkan bahwa TDNN sangat baik untuk pembangunan model dalam ranah suara.

4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan server milik Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Syiah Kuala dengan spesifikasi prosesor Intel Xeon Gold 5218, CPU 2.30 GHz 64 inti, RAM 128 GB, dan penyimpanan sebesar 8 TB. Server tersebut memiliki empat slot Graphics Processing Unit (GPU) dari Video Graphics Adapter (VGA) NVIDIA GeForce RTX 2080 Ti dengan VRAM 12 GB. Setelah semua model berhasil dibangun, Model-I berukuran 33 MB, Model-II berukuran 12 MB, Model-III berukuran 32 MB, dan Model-IV berukuran 11 MB.

Berdasarkan ukuran model yang didapat, arsitektur DNN kedua menghasilkan model berukuran yang lebih kecil dibandingkan model yang dibangun menggunakan arsitektur DNN pertama dan konfigurasi MFCC kedua menghasilkan model berukuran lebih kecil dibandingkan model yang dibangun menggunakan konfigurasi MFCC pertama. Hal ini disebabkan oleh arsitektur DNN kedua memiliki jumlah node yang lebih sedikit dibandingkan arsitektur DNN pertama sehingga jumlah bobot yang disimpan lebih sedikit. Akurasi model untuk setiap data testing dapat dilihat pada Tabel 6. Durasi ekstraksi x-vectors untuk setiap data test dapat dilihat pada Tabel 7.

Berdasarkan Tabel 6, model yang memiliki akurasi tertinggi untuk pengujian menggunakan dataset VoxCeleb1 bagian test adalah Model-I, sedangkan Model-II menempati urutan kedua dengan selisih EER sebesar 0,36%. Dengan demikian, hanya Model-I yang memiliki akurasi yang lebih tinggi dibandingkan model yang dijadikan baseline penelitian ini, yaitu penelitian Xie pada tahun 2019 dengan selisih nilai EER sebesar 0,07%. Pada pengujian menggunakan dataset inf19_test_td, didapati bahwa Model-IV memiliki akurasi tertinggi, sedangkan Model-II menempati urutan kedua dengan selisih EER sebesar 0,07%. Pengujian menggunakan dataset inf19_test_tid menunjukkan bahwa Model-II memiliki akurasi tertinggi, sedangkan Model-IV memiliki akurasi kedua tertinggi dengan selisih EER sebesar 0,54%.

Berdasarkan Tabel 7, dapat dilihat bahwa model yang memiliki durasi paling singkat untuk mengekstrak x-vectors dari setiap dataset adalah Model-IV, sedangkan Model-II menempati urutan kedua dengan selisih durasi total sebesar 31 menit 1 detik. Berdasarkan analisis hasil pengujian model, Model-II dan Model-IV, yang dibangun dengan menggunakan arsitektur DNN kedua, memiliki akurasi rata-rata terbaik dibandingkan dengan Model-I dan Model-III. Dengan adanya pengurangan jumlah node, arsitektur DNN kedua ini lebih cocok untuk dataset INF19 sehingga menghasilkan akurasi yang jauh lebih baik dibandingkan akurasi yang dihasilkan oleh model yang dibangun menggunakan arsitektur DNN pertama. Hal ini dikarenakan bobot yang disimpan pada model yang dibangun menggunakan arsitektur DNN kedua lebih akurat dalam merepresentasikan fitur dari dataset INF19, yaitu dataset audio yang dibangun dengan menggunakan suara pembicara berbahasa Indonesia. Model yang dibangun menggunakan konfigurasi MFCC kedua dan arsitektur DNN kedua dapat membuat durasi ekstraksi x-vectors lebih singkat dibandingkan model yang dibangun menggunakan konfigurasi MFCC pertama dan arsitektur DNN pertama. Hal ini dikarenakan semakin kecil dimensi fitur dan semakin kecil jumlah node pada DNN, semakin singkat durasi pemrosesan menggunakan model tersebut. Secara keseluruhan, Model-II adalah model yang terbaik. Dibandingkan Model-IV, akurasi Model-II lebih unggul terutama untuk dua dari tiga dataset yang telah diujikan. Meskipun penggunaan Model-IV dapat mempersingkat durasi ekstraksi X-Vector, namun selisih durasinya sangat kecil dan dampaknya tidak terlalu signifikan. Pemilihan ini juga didukung oleh selisih yang dekat antara Model-II ketika menempati urutan kedua terbaik dengan model terbaik pada pengujian akurasi dan durasi. Selain itu, penelitian ini berfokus pada penerapan model speaker identification untuk pembicara berbahasa Indonesia.

Tabel 6. Akurasi Model

Nama Model	VoxCeleb1 *test*		inf19_test_td		inf19_test_tid	
	EER	***Threshold***	**EER**	***Threshold***	**EER**	***Threshold***
Model-I	**3,15%**	-3,28496	14,93%	-615,08	10,63%	-409,022
Model-II	3,51%	-2,4826	1,3%	-30,876	**1,4%**	-17,2181
Model-III	4,21%	-2,69932	16,71%	-593,319	12,4%	-388,591
Model-IV	4,52%	-1,74275	**1,23%**	-32,3137	1,94%	-20,1383

Tabel 7. Durasi Ekstraksi X-Vectors

Nama Model	Durasi (jam:menit:detik)					
	Data train	**VoxCeleb1 *test***	**inf19_enroll**	**inf19_test_td**	**inf19_test_tid**	**Total**
Model-I	7:5:54	0:2:36	0:0:15	0:0:25	0:0:9	7:8:49
Model-II	6:42:39	0:2:24	0:0:18	0:0:25	0:0:9	6:45:55
Model-III	7:16:7	0:2:37	0:0:14	0:0:24	0:0:9	7:19:31
Model-IV	**6:11:53**	**0:2:18**	**0:0:13**	**0:0:22**	**0:0:8**	**6:14:54**

5. KESIMPULAN

Penelitian ini berhasil membangun dataset INF19 yang merupakan dataset audio dari pembicara berbahasa Indonesia. Dataset yang dibangun menggunakan rekaman suara mahasiswa Jurusan Informatika USK angkatan 2019 berjumlah 2.160 berkas audio dari 72 pembicara dengan durasi total sekitar 3 jam 30 menit. Berdasarkan penelitian yang telah dilakukan, arsitektur DNN dengan memanfaatkan TDNN dapat menghasilkan akurasi yang baik. Selain itu, ukuran model juga dapat diperkecil dan durasi ekstraksi x-vectors dapat dipersingkat dengan cara mengurangi jumlah node pada lapisan DNN dan mengecilkan dimensi fitur MFCC. Untuk ke depan, penelitian dapat dilakukan pada penggunaan arsitektur yang berbeda dengan menggunakan data aset yang sama sehingga bisa dilakukan perbandingan hasil yang didapatkan.

DAFTAR PUSTAKA

BAI, Z., & ZHANG, X.-L., 2021. Speaker Recognition Based on Deep Learning: An Overview. Neural Networks, 140, 65–99. https://doi.org/10.1016/j.neunet.2021.03.004

CAO, C., LIU, F., TAN, H., SONG, D., SHU, W., LI, W., ZHOU, Y., BO, X., & XIE, Z., 2018. Deep Learning and Its Applications In Biomedicine. Genomics, Proteomics & Bioinformatics, 16(1), 17–32. https://doi.org/10.1016/j.gpb.2017.07.003

CHUNG, J. S., NAGRANI, A., & ZISSERMAN, A., 2018. Voxceleb2: Deep Speaker Recognition. arXiv preprint arXiv:1806.05622. https://doi.org/10.21437/Interspeech.2018-1929

DING, S., CHEN, T., GONG, X., ZHA, W., & WANG, Z., 2020. Autospeech: Neural Architecture Search for Speaker Recognition. arXiv preprint arXiv:2005.03215. https://doi.org/10.21437/Interspeech.2020-1258

FAN, C., LIU, B., TAO, J., YI, J., WEN, Z., & SONG, L., 2021. Deep Time Delay Neural Network for Speech Enhancement with Full Data Learning. 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP), 1–5. https://doi.org/10.1109/ISCSLP49672.2021.9362059

FURUI, S., 2018. Digital Speech Processing: Synthesis, and Recognition. CRC Press. https://doi.org/10.1201/9781482270648

MISBULLAH, A., NAZARUDDIN, N., MARZUKI, M., & ZULFAN, Z., 2020. Penerapan Time Delay Neural Network pada Model Akustik untuk Sistem Voice-to-Text Berbahasa Sunda. Journal of Data Analysis, 2(2), 61–70. https://doi.org/10.24815/jda.v2i2.15235

NURSHOLIHATUN, E., 2020. Identifikasi Suara Menggunakan Metode Mel Frequency Cepstrum Coefficients (MFCC) dan Jaringan Syaraf Tiruan Backpropagation. Universitas Mataram. https://dielektrika.unram.ac.id/index.php/dielektrika/article/view/232

PEDDINTI, V., POVEY, D., & KHUDANPUR, S., 2015. A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2015-Janua, 3214–3218. https://doi.org/10.21437/interspeech.2015-647

POVEY, D., GHOSHAL, A., BOULIANNE, G., BURGET, L., GLEMBEK, O., GOEL, N., HANNEMANN, M., MOTLICEK, P., QIAN, Y., SCHWARZ, P., SILOVSKY, J., STEMMER, G., & VESELY, K., 2011. The Kaldi Speech Recognition Toolkit. IEEE Signal Processing Society. https://infoscience.epfl.ch/record/192584

SNYDER, D., GARCIA-ROMERO, D., SELL, G., POVEY, D., & KHUDANPUR, S., 2018. X-Vectors: Robust DNN Embeddings For Speaker Recognition. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), 5329–5333. https://doi.org/10.1109/ICASSP.2018.8461375

XIANG, X., WANG, S., HUANG, H., QIAN, Y., & YU, K., 2019. Margin Matters: Towards More Discriminative Deep Neural Network Embeddings For Speaker Recognition. 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 1652–1656. https://doi.org/10.1109/APSIPAASC47483.2019.9023039

XIE, W., NAGRANI, A., CHUNG, J. S., & ZISSERMAN, A., 2019. Utterance-Level Aggregation for Speaker Recognition In The Wild. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5791–5795. https://doi.org/10.1109/ICASSP.2019.8683120