Variasi Implementasi Model *Speaker Recognition* dalam Mendeteksi Kehadiran Mahasiswa

Speech Recognition

Kelompok: 18

Nama-nama dan NIM anggota kelompok
- Fendy Wijaya (2602092150)
- Leonardo Dahendra (2602097076)
- Wilbert Yang (2602093802)

I. Latar Belakang

Suara merupakan suatu biometrik yang menyimpan informasi terkait sifat yang dimiliki oleh seseorang, misalnya etnis, umur, jenis kelamin, dan perasaan dari pembicara (Hanifa, Isa, & Mohamad, 2021). Seiring dengan kemajuan zaman, produk teknologi di bidang pengenalan suara berkembang dengan sangat pesat di berbagai aspek kehidupan. Salah satu bidang yang menjadi cabang pada ilmu pengenalan suara adalah *speaker recognition*. *Speaker recognition* adalah suatu teknik untuk mengidentifikasi seseorang berdasarkan suaranya (Bai & Zhang, 2021). Dengan adanya *speaker recognition*, suatu sistem dapat mengenali siapa pemilik suatu suara yang diberikan kepadanya.

Dalam lingkungan akademis seperti di kampus, penerapan teknologi *speaker recognition* juga mampu menjadi sarana peningkatan mutu kampus. Salah satu masalah umum yang mengundang perhatian khalayak adalah metode pengabsenan mahasiswa yang terstruktur dan sistematis (Lazam & Saparon, 2021). Kampus perlu memastikan kedisiplinan mahasiswa melalui pengabsenan yang efektif dan tepat. Dengan demikian, pendataan kehadiran dapat dilakukan tanpa adanya kecurangan-kecurangan yang mungkin dilakukan oleh mahasiswa, misalnya fenomena titip absen atau pengaksesan akun kampus mahasiswa lain untuk didatakan hadir.

Prinsip *speaker recognition* menjadi salah satu metode yang tepat untuk melakukan pengabsenan terhadap mahasiswa secara komprehensif dan tepat sasaran. Penggunaan suara dapat mengantisipasi kejadian-kejadian di mana *fingerprint* mengalami kegagalan fungsi atau kerja akibat gejala fisik, misalnya luka pada jari atau amputasi jari. Selain itu, setiap orang memiliki jenis dan warna suara yang unik. Hal ini bermakna bahwa biometrik suara mampu membedakan satu orang dengan orang lainnya (Uddin et al, 2016). Prinsip inilah yang dapat dimanfaatkan untuk mengimplementasikan sistem pendataan kehadiran dengan meminimalisasi kemungkinan kecurangan titip absen oleh mahasiswa.

Teknologi *speaker recognition* rupanya menyimpan potensi dalam hal penegakan kedisiplinan di kehidupan nyata. Oleh karena itu, proyek ini akan membangun suatu sistem *speaker recognition* yang mampu mengenali suara-suara mahasiswa. Keluaran proyek ini berupa sistem yang mampu mendeteksi nama mahasiswa yang memberi input suara kepada sistem. Harapannya, keluaran ini menjadi fondasi awal untuk membangun sebuah aplikasi terkait di penelitian-penelitian mendatang.

II. Rumusan Masalah

Rumusan masalah yang ingin diselesaikan pada *project* ini dapat dirincikan sebagai berikut.

- Bagaimana cara membangun model yang mampu mengenali *speaker* dari suatu suara?
- Bagaimanakah durasi *dataset* yang baik untuk menghasilkan performa terbaik dari model RNN, SVM, dan XGBoost?
- Model apakah yang memiliki performa terbaik dalam menjalankan *speaker recognition*?

III. Tujuan Project

Tujuan yang ingin dicapai pada *project* ini adalah sebagai berikut.

- Memahami cara membangun model yang mampu mengenali *speaker* dari suatu suara.
- Memperkirakan durasi *dataset* yang baik untuk menghasilkan performa terbaik dari model RNN, SVM, dan XGBoost.
- Menganalisis model yang memiliki performa terbaik dalam menjalankan *speaker recognition*.

IV. Ruang Lingkup Project

Dalam pengerjaan *project* ini, masalah dibatasi hanya pada ruang lingkup mahasiswa tertentu saja. *Project* ini melibatkan empat orang mahasiswa yang suaranya akan dilibatkan dalam pembuatan *dataset*, sehingga model hanya mampu memprediksi dengan *range* keempat nama mahasiswa tersebut saja. Selain itu, terdapat tambahan suara manusia yang diambil langsung dari Kaggle sebanyak tiga orang, sehingga total jumlah *range* prediksi hanya berada pada tujuh nama tersebut saja.

V. Deskripsi Dataset

Dataset dibangun dengan menggabungkan data suara mahasiswa dan data suara dari Kaggle. Data suara yang diambil sebanyak 4 audio dari 4 mahasiswa berbeda, serta 3 audio dari 3 *speaker* berbeda yang bersumber dari Kaggle dengan tautan terlampir. Untuk jumlah audio setiap speaker, terdapat sedikit ketidakseimbangan dengan pembagian durasi audio setiap speaker secara berurutan sebesar 37.9, 48, 30, 11.9, 17.7, 12.3, dan 8.5 menit.

VI. Tahap-tahap Eksperimen

Secara garis besar, eksperimen yang dilaksanakan dalam *project* ini melibatkan beberapa tahapan sebagai berikut.

(1) Data Gathering

Data diambil dari Kaggle dan direkam dari suara manusia asli sesuai yang telah disebutkan sebelumnya. Kemudian, data audio yang bersumber dari Kaggle dipilah terlebih dahulu dan dikonversi ke dalam format .wav.

(2) Data Preprocessing

Audio dipotong-potong menjadi beberapa bagian dengan durasi yang sama. Kemudian, audio-audio tersebut akan diekstrak fitur-fitur MFCC dan label-labelnya. *Output* yang dihasilkan akan dikodekan ke dalam bentuk numerik. Lalu, *dataset* akan di-*split* untuk proses *training, testing,* dan *validation* (khusus untuk model *deep learning*), atau hanya *training* dan *testing* saja (khusus model *machine learning*).

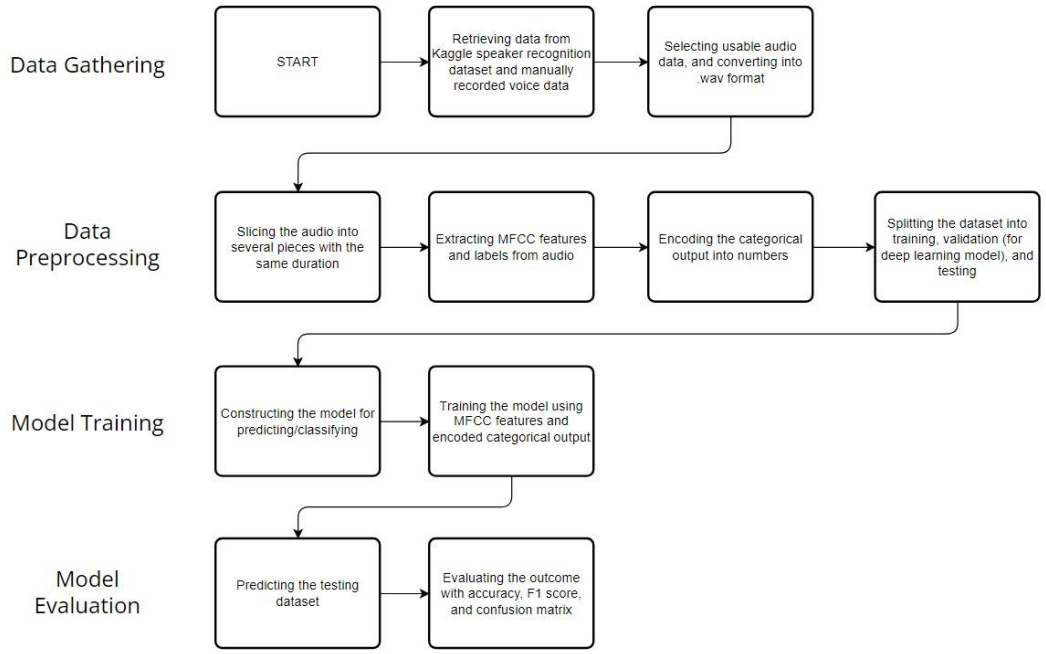
(3) Model Training

Model dibangun serta dilatih dengan menggunakan fitur MFCC dan *output* kategorik yang telah di-*encode*.

(4) Model Evaluation

Testing dataset diujikan kepada model dan hasilnya akan dievaluasi melalui metrik-metrik seperti akurasi, F1, dan *confusion matrix*.

Tahapan-tahapan tersebut juga dapat dideskripsikan melalui diagram alir berikut.



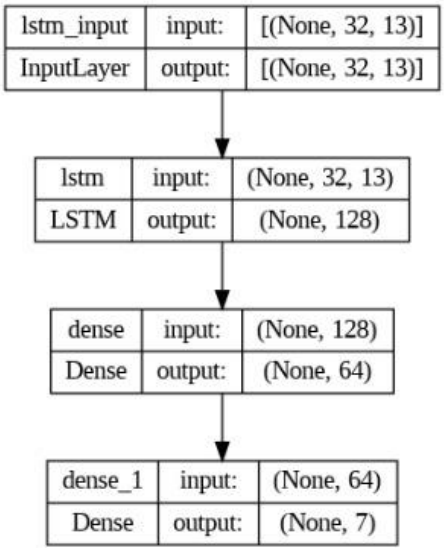
VII. Preprocessing

Setelah memperoleh data audio yang diinginkan, proses selanjutnya adalah *slicing* (pemotongan) audio menjadi sampel-sampel audio per 2, 5, dan 20 detik, serta pembuangan sisa audio yang durasinya di bawah angka tersebut. Setiap audio besar tersebut dipotong-potong sehingga dihasilkan 4.994, 1.998, dan 496 sampel audio berbeda yang terbagi untuk 7 *speaker*. Setiap audio ini kemudian akan diekstraksi MFCC *features*-nya, yang akan di-*feed* ke dalam model. Langkah selanjutnya adalah proses *encoding* terhadap label untuk dikonversi menjadi angka. Terakhir, *dataset* akan dibagi menjadi 3 bagian (untuk model RNN), dengan komposisi 70% *training*, 15% *validation*, dan 15% *testing*. Sementara untuk model SVM dan XGBoost, *dataset* dibagi menjadi 2 bagian dengan komposisi 80% *training* dan 20% *testing*.

VIII. Rancangan Model

Sebagai bentuk perbandingan, *project* ini mengusung tiga model, yaitu *Support Vector Machine Classifier* (SVM), XGBoost, dan *Recurrent Neural Network* (RNN). Di antara ketiga model ini, SVM dan XGBoost berada dalam bidang *machine learning*, sementara RNN berada dalam bidang *deep learning*. Untuk model-model *machine learning*, seperti SVM dan XGBoost, model dibangun dari *library* yang sudah ada, lalu di-*training* dari awal dengan menggunakan *dataset* yang sudah dipersiapkan.

Sementara itu, model RNN dibangun dari 4 layer, yang tersusun atas 1 *input layer*, 2 *hidden layer*, dan 1 *output layer*. *Input layer* dibentuk dengan ukuran 32 x 13, yang ekuivalen dengan ukuran MFCC dari satu data audio. Selanjutnya, *hidden layer* terdiri atas LSTM layer dengan ukuran 128 neuron dan Dense layer dengan ukuran 64 neuron, yang masing-masing diaktivasi oleh ReLU (*Rectified Linear Unit*) *activation function*. Terakhir, *output layer* dibentuk oleh Dense layer dengan ukuran 7 neuron, sesuai dengan jumlah label berbeda dalam *dataset,* yang diaktivasi oleh Softmax *activation function*. Secara umum, berikut ini adalah gambaran arsitektur model RNN yang diusulkan.



IX. Proses Pelatihan Model

Pada model SVM, ada beberapa parameter yang diubah terlebih dahulu. *Kernel* yang digunakan pada model ini adalah 'rbf' (*radial basis function*) agar model dapat menangani relasi non-linear antardata. Selain itu, *regularization parameter* diatur ke nilai 1, dan nilai *gamma* menggunakan *scale* agar model otomatis menyesuaikan dengan fitur-fitur pada *dataset*. Sementara model *machine learning* lainnya, XGBoost, menggunakan parameter *default*-nya saja untuk menjalankan proses *training*.

Beralih ke model RNN, kami mengimplementasikan *optimizer* Adam yang paling sering digunakan, dan untuk *loss*, kami menggunakan *Sparse Categorical Crossentropy* yang cocok untuk *multiclass* dengan label berbentuk indeks dan bukan bersifat *one-hot*. Selain itu, implementasi *early stopping* juga sangat diperlukan di sini agar model segera berhenti melakukan *training* jika *validation loss* terus mengalami kenaikan setelah melalui dua iterasi. Kami juga akan menggunakan *restore_best_weight* agar model akhir kami mengembalikan model dengan akurasi terbaik ketika *training*. Model ini akan di-*training* selama 20 *epochs* dengan *batch size* sebesar 32.

X. Hasil Evaluasi dan Analisis

Model berbasis *machine learning* yang telah dibangun dan dilatih akan dievaluasi terhadap ketiga jenis *dataset* yang dipakai, yaitu *dataset* berdurasi 2 detik, 5 detik, dan 20 detik. Hasil evaluasi dapat dideskripsikan melalui nilai akurasi model yang ditampilkan melalui tabel perbandingan sebagai berikut.

Model ML	*Dataset* 2 detik	*Dataset* 5 detik	*Dataset* 20 detik
SVM	97%	96,99%	72%
XGBoost	98%	96,89%	84%

Dari perbandingan yang dilakukan terhadap kedua model berbasis *machine learning* tersebut, terlihat bahwa *dataset* dengan potongan audio sebesar 2 detik selalu menghasilkan nilai akurasi terbesar. Jika dibandingkan, model XGBoost adalah model terbaik yang menghasilkan akurasi mencapai 98% pada *dataset* 2 detik. Nilai akurasi ini terpaut 1% dengan SVM yang hanya mencapai 97% pada *dataset* yang sama.

Model RNN	*Dataset* 2 detik	*Dataset* 5 detik	*Dataset* 20 detik
Training accuracy	98,28%	96,88%	86,74%
Validation accuracy	99%	97,46%	93,24%
Testing accuracy	95,66%	98,26%	90,66%
Jumlah *epochs*	13	6	11

Selanjutnya untuk model RNN, dari hasil *training* yang dilakukan, terdapat tren yang cukup jelas, yaitu akurasi model pada *dataset* berdurasi 20 detik pasti lebih rendah daripada *dataset* berdurasi 5 detik atau 2 detik. Hal ini kemungkinan besar terjadi karena kecilnya *dataset* yang ada, dengan data setiap *speaker* tidak menyentuh 1 jam, sehingga ketika dibagi menjadi 20 detik, jumlah data yang ada menjadi sangat sedikit dan model akan kesulitan dalam membaca pola-pola data yang ada. Akan tetapi, *dataset* 2 detik dan 5 detik masih menunjukkan performa yang sangat baik dan signifikan, dengan nilai yang tidak berbeda jauh dengan akurasi pada model *machine learning*. Untuk model berbasis RNN, nilai terbaik berada pada *dataset* 5 detik, dengan nilai akurasi *testing* yang lebih besar 0,26% dari akurasi terbaik pada model *machine learning* yang dicapai oleh XGBoost sebesar 98%.

XI. Kesimpulan dan Saran

Secara garis besar, *speaker recognition* dapat diimplementasikan pada sebuah model dengan membagi audio menjadi potongan-potongan kecil berdurasi pendek yang akan diekstraksi menjadi fitur-fitur MFCC. Durasi inilah yang perlu disesuaikan dengan model yang digunakan agar memperoleh akurasi maksimal. Berdasarkan hasil eksperimen, kombinasi model dan durasi sampel audio yang optimal adalah RNN dengan durasi sampel 5 detik, yang menghasilkan akurasi 98,26%. Akan tetapi, jika patokan awalnya adalah *machine learning*, maka kombinasi yang optimal adalah XGBoost dengan durasi sampel 2 detik, yang akurasinya tidak jauh berbeda, yaitu 98%.

Sepanjang hasil eksperimen yang telah dilaksanakan pada *project* ini, model yang dibangun telah menunjukkan performa yang sangat baik dan mengarah pada satu simpulan utama yang jelas. Akan tetapi, untuk membangun sebuah sistem yang ditujukan bagi pendeteksi kehadiran mahasiswa, diperlukan suatu sistem yang mampu mendeteksi audio secara *real-time*. Keluaran dari *project* ini hanya berupa sistem *artificial intelligence* yang sudah mampu memprediksi *speaker* yang sedang berbicara, belum berupa pendeteksi *real-time* secara langsung untuk mengabsen mahasiswa. Sistem ini masih perlu dikembangkan lebih lanjut secara implementatif pada suatu aplikasi agar dapat dimanfaatkan oleh institusi seperti perguruan tinggi. Harapannya, eksperimen dan *project* atau penelitian lain di masa yang akan datang mampu meningkatkan kematangan sistem agar dapat dimanfaatkan secara langsung sebagai sebuah pendeteksi kehadiran mahasiswa.

XII. Lampiran

Notebook 2-second-dataset:
https://colab.research.google.com/drive/1itR0DvmrnlDi2LegbN0VC-S1YiJQxa6h?usp=sharing

Notebook 5-second-dataset:
https://colab.research.google.com/drive/1qQLBKT1u_N-F-jfozZPk1nmsjLFWcRLq?usp=sharing

Notebook 20-second-dataset:
https://colab.research.google.com/drive/1cDmHJDfc6vmRBIAQqoqBI4vpbVwAEMF2?usp=sharing

Dataset Kaggle:
https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset

Audio mahasiswa dan audio *speaker* Kaggle:
https://drive.google.com/drive/folders/17BcrSTngwZzOqeQHdhcQrPD29w_P2pMb

Daftar Pustaka

Bai, Z., & Zhang, X. L. (2021). Speaker recognition based on deep learning: An overview. *Neural Networks*, 140, 65-99.

Hanifa, R. M., Isa, K., & Mohamad, S. (2021). A review on speaker recognition: Technology and challenges. *Computers & Electrical Engineering, 90*, 107005.

Lazam, N. A. M., & Saparon, A. (2021). Development of academic attendance system using voice verification. *Malaysian Journal of Computer Science*, 106-120.

Uddin, N., Rashid, M. M., Mostafa, M. G., Belayet, H., Salam, S. M., Nithe, N. A., & Halder, S. (2016). Development of voice recognition for student attendance. *Global Journal of Human-Social Science Research: G Linguistics & Education, 16*(1), 1-6.

Feedback

Belum semua arsitektur model yang digunakan disampaikan dalam laporan. Perjelas secara eksplisit mengenai jumlah kelas pada eksperimen Anda. Akan lebih baik jika hasil evaluasi model dilengkapi dengan confusion matrix. Analisis hasil evaluasi model mengenai alasan akurasi data durasi lebih panjang lebih rendah dibanding data dengan durasi lebih singkat belum disampaikan.

UAS SPEECH RECOGNITION

1. DESKRIPSI PROYEK

➢ Latar Belakang

Suara merupakan suatu biometrik yang menyimpan informasi terkait sifat yang dimiliki oleh seseorang, misalnya etnis, umur, jenis kelamin, dan perasaan dari pembicara (Hanifa, Isa, & Mohamad, 2021). Seiring dengan kemajuan zaman, produk teknologi di bidang pengenalan suara berkembang dengan sangat pesat di berbagai aspek kehidupan. Salah satu bidang yang menjadi cabang pada ilmu pengenalan suara adalah *speaker recognition*. *Speaker recognition* adalah suatu teknik untuk mengidentifikasi seseorang berdasarkan suaranya (Bai & Zhang, 2021). Dengan adanya *speaker recognition*, suatu sistem dapat mengenali siapa pemilik suatu suara yang diberikan kepadanya.

Dalam lingkungan akademis seperti di kampus, penerapan teknologi *speaker recognition* juga mampu menjadi sarana peningkatan mutu kampus. Salah satu masalah umum yang mengundang perhatian khalayak adalah metode pengabsenan mahasiswa yang terstruktur dan sistematis (Lazam & Saparon, 2021). Kampus perlu memastikan kedisiplinan mahasiswa melalui pengabsenan yang efektif dan tepat. Dengan demikian, pendataan kehadiran dapat dilakukan tanpa adanya kecurangan-kecurangan yang mungkin dilakukan oleh mahasiswa, misalnya fenomena titip absen atau pengaksesan akun kampus mahasiswa lain untuk didatakan hadir.

Prinsip *speaker recognition* menjadi salah satu metode yang tepat untuk melakukan pengabsenan terhadap mahasiswa secara komprehensif dan tepat sasaran. Penggunaan suara dapat mengantisipasi kejadian-kejadian di mana *fingerprint* mengalami kegagalan fungsi atau kerja akibat gejala fisik, misalnya luka pada jari atau amputasi jari. Selain itu, setiap orang memiliki jenis dan warna suara yang unik. Hal ini bermakna bahwa biometrik suara mampu membedakan satu orang dengan orang lainnya (Uddin et al, 2016). Prinsip inilah yang dapat dimanfaatkan untuk mengimplementasikan sistem pendataan kehadiran dengan meminimalisasi kemungkinan kecurangan titip absen oleh mahasiswa.

Teknologi *speaker recognition* rupanya menyimpan potensi dalam hal penegakan kedisiplinan di kehidupan nyata. Oleh karena itu, proyek ini akan membangun suatu sistem *speaker recognition* yang mampu mengenali suara-suara mahasiswa. Keluaran proyek ini berupa sistem yang mampu mendeteksi nama mahasiswa yang memberi input suara kepada sistem. Harapannya, keluaran ini menjadi fondasi awal untuk membangun sebuah aplikasi terkait di penelitian-penelitian mendatang.

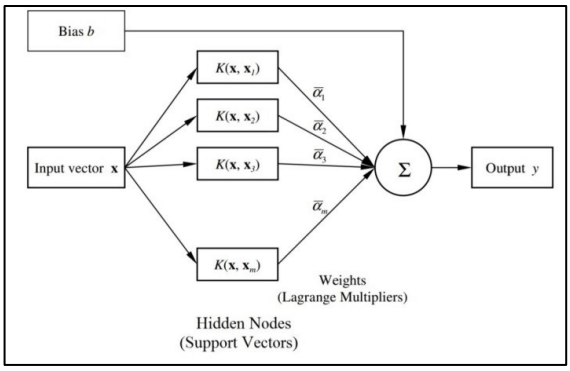
➢ Masalah yang Diselesaikan

Masalah yang ingin diselesaikan pada *project* ini dapat dirincikan sebagai berikut.

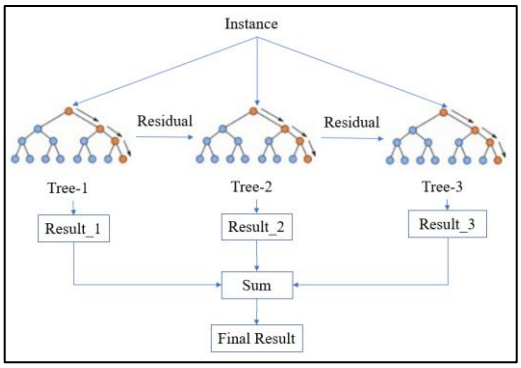
- Maraknya fenomena titip absen yang mudah diakali oleh mahasiswa
- Kebutuhan membangun model yang mampu mengenali *speaker* dari suatu suara
- Minimnya pengetahuan mengenai durasi *dataset* yang baik untuk menghasilkan performa terbaik dari model RNN, SVM, dan XGBoost
- Kebutuhan untuk memahami model yang memiliki performa terbaik dalam menjalankan *speaker recognition*

➢ **Metode yang Diusulkan**

Sebagai bentuk perbandingan, *project* ini mengusung tiga model, yaitu *Support Vector Machine Classifier* (SVM), XGBoost, dan *Recurrent Neural Network* (RNN). Di antara ketiga model ini, SVM dan XGBoost berada dalam bidang *machine learning*, sementara RNN berada dalam bidang *deep learning*. Untuk model-model *machine learning*, seperti SVM dan XGBoost, model dibangun dari *library* yang sudah ada, lalu di-*training* dari awal dengan menggunakan *dataset* yang sudah dipersiapkan. Gambaran arsitektur model SVM dan XGBoost ditunjukkan pada visual berikut.

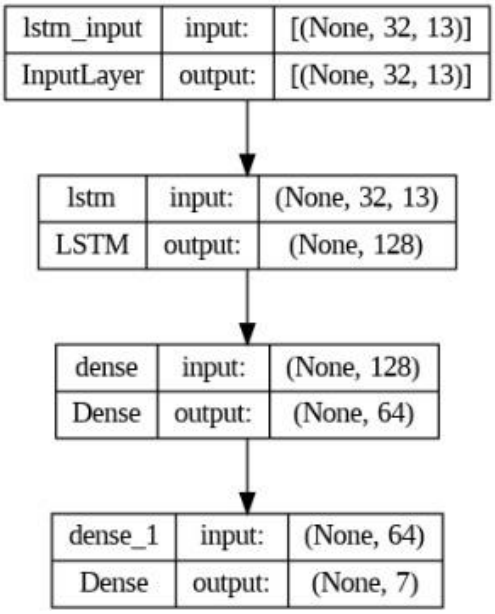


SVM



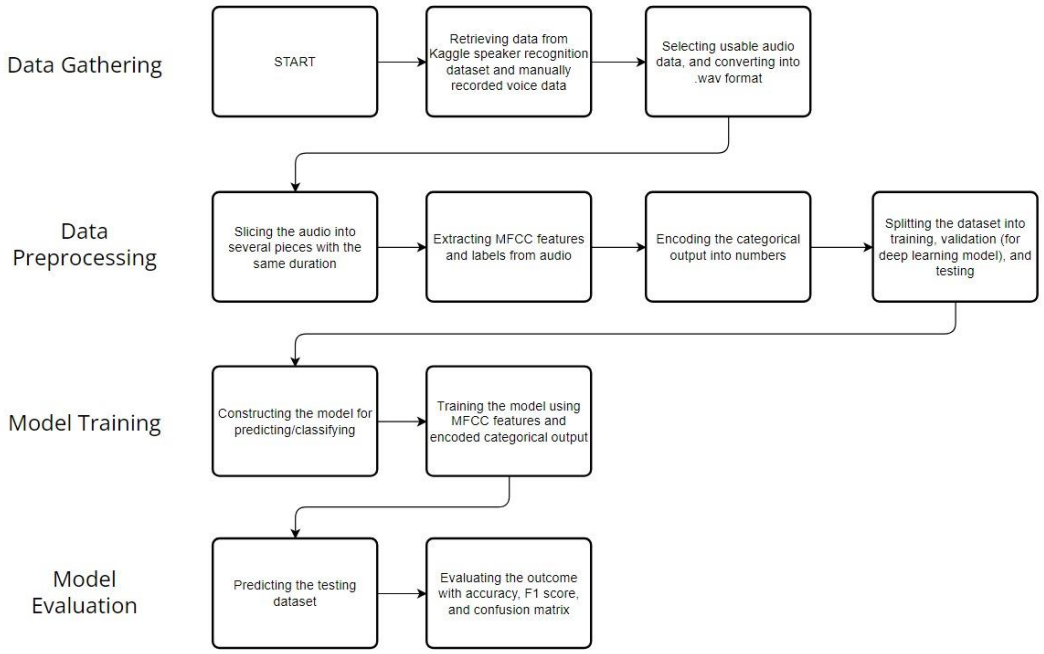
XGBoost

Sementara itu, model RNN dibangun dari 4 layer, yang tersusun atas 1 *input layer*, 2 *hidden layer*, dan 1 *output layer*. *Input layer* dibentuk dengan ukuran 32 x 13, yang ekuivalen dengan ukuran MFCC dari satu data audio. Selanjutnya, *hidden layer* terdiri atas LSTM layer dengan ukuran 128 neuron dan Dense layer dengan ukuran 64 neuron, yang masing-masing diaktivasi oleh ReLU (*Rectified Linear Unit*) *activation function*. Terakhir, *output layer* dibentuk oleh Dense layer dengan ukuran 7 neuron, sesuai dengan jumlah label berbeda dalam *dataset,* yang diaktivasi oleh Softmax *activation function*. Secara umum, berikut ini adalah gambaran arsitektur model RNN yang diusulkan.



RNN

2. DIAGRAM ALIR INFORMASI DARI INPUT KE OUTPUT



3. ALUR EKSPERIMEN

- **Dataset & Preprocessing**

Dataset dibangun dengan menggabungkan data suara mahasiswa dan data suara dari Kaggle. Data suara yang diambil sebanyak 4 audio dari 4 mahasiswa berbeda, serta 3 audio dari 3 *speaker* berbeda yang bersumber dari Kaggle. Untuk jumlah audio setiap speaker, terdapat sedikit ketidakseimbangan dengan pembagian durasi audio setiap speaker secara berurutan sebesar 37.9, 48, 30, 11.9, 17.7, 12.3, dan 8.5 menit. Dengan demikian, secara keseluruhan, total jumlah kelas pada eksperimen ini adalah sebanyak 7 kelas.

Setelah memperoleh data audio yang diinginkan, proses selanjutnya adalah slicing (pemotongan) audio menjadi sampel-sampel audio per 2, 5, dan 20 detik, serta pembuangan sisa audio yang durasinya di bawah angka tersebut. Setiap audio besar tersebut dipotong-potong sehingga dihasilkan 4.994, 1.998, dan 496 sampel audio berbeda yang terbagi untuk 7 speaker. Setiap audio ini kemudian akan diekstraksi MFCC features-nya, yang akan di-feed ke dalam model. Langkah selanjutnya adalah proses encoding terhadap label untuk dikonversi menjadi angka.

- **Pembagian data latih dan uji**

Dataset akan dibagi menjadi 3 bagian (untuk model RNN), dengan komposisi 70% training, 15% validation, dan 15% testing. Sementara untuk model SVM dan XGBoost, dataset dibagi menjadi 2 bagian dengan komposisi 80% training dan 20% testing. Pembagian *dataset* ini dilakukan dengan menggunakan bantuan library scikit-learn.

- **Proses training (tuning)**

Untuk model *machine learning* (SVM dan XGBoost), dari 80% data *training* yang telah dibentuk sebelumnya, data-data tersebut kemudian akan di-*feed* untuk pelatihan kedua model. Pada model SVM, ada beberapa parameter yang diubah terlebih dahulu. *Kernel* yang digunakan pada model ini adalah 'rbf' (*radial basis function*) agar model dapat menangani relasi non-linear antardata. Selain itu, *regularization parameter* diatur ke nilai 1, dan nilai *gamma* menggunakan *scale* agar model otomatis menyesuaikan dengan fitur-fitur pada *dataset*. Sementara model *machine learning* lainnya, XGBoost, menggunakan parameter *default*-nya saja untuk menjalankan proses *training*.

Beralih ke model RNN, 70% data *training* dan 15% data *validation* akan dipakai untuk menjalankan proses *training* model. Kami mengimplementasikan *optimizer* Adam yang paling sering digunakan, dan untuk *loss*, kami menggunakan *Sparse Categorical Crossentropy* yang cocok untuk *multiclass* dengan label berbentuk indeks dan bukan bersifat *one-hot*. Selain itu, implementasi *early stopping* juga sangat diperlukan di sini agar model segera berhenti melakukan *training* jika *validation loss* terus mengalami kenaikan setelah melalui dua iterasi. Kami juga akan menggunakan *restore_best_weight* agar model akhir kami mengembalikan model dengan akurasi terbaik ketika *training*. Model ini akan di-*training* selama 20 *epochs* dengan *batch size* sebesar 32.

- **Pengujian model**

Setelah proses *training* selesai, baik untuk model SVM, XGBoost, dan RNN, ketiganya akan dilakukan proses pengujian dengan menggunakan data *testing* hasil pembagian *dataset* yang telah dilakukan sebelumnya. Dalam hal ini, untuk model berbasis *machine learning*, pengujian menggunakan 20% data uji yang tersisa. Sedangkan untuk model berbasis *deep learning*, pengujian akan menggunakan 15% data uji yang tersisa. Pengujian model ini dilakukan untuk menguji kemampuan prediksi model berdasarkan pola-pola data yang telah diperoleh dari hasil pelatihan sebelumnya.

- **Cara evaluasi model**

Hasil pengujian untuk masing-masing model akan dievaluasi melalui metrik-metrik seperti akurasi, F1-score, dan *confusion matrix*. Metrik-metrik tersebut tersedia dalam library scikit-learn dan dapat digunakan dengan mudah untuk menghitung besaran evaluasi tersebut sesuai hasil pengujian model yang telah dilakukan pada tahap sebelumnya.

4. HASIL EVALUASI (MINIMAL 2 SKENARIO)

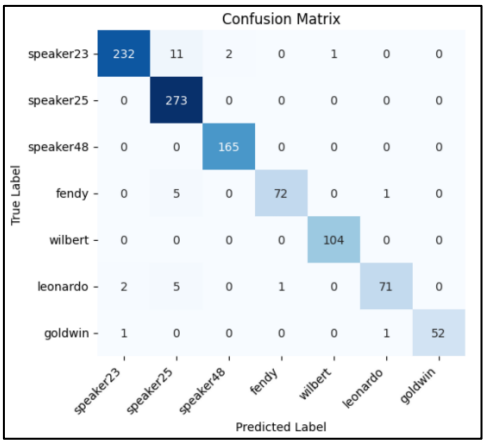
Model berbasis *machine learning* yang telah dibangun dan dilatih akan dievaluasi terhadap ketiga jenis *dataset* yang dipakai, yaitu *dataset* berdurasi 2 detik, 5 detik, dan 20 detik. Hasil evaluasi dapat dideskripsikan melalui nilai akurasi model yang ditampilkan melalui tabel perbandingan sebagai berikut.

Model ML	***Dataset* 2 detik**	***Dataset* 5 detik**	***Dataset* 20 detik**
SVM	97%	96,99%	72%
XGBoost	98%	96,89%	84%

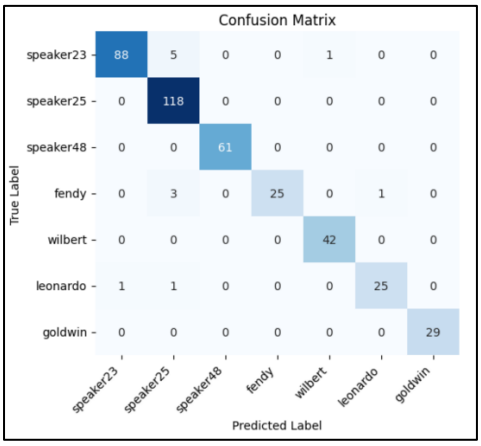
Sementara perolehan F1-score pada model-model *machine learning* disajikan pada tabel berikut.

Model ML	*Dataset* 2 detik	*Dataset* 5 detik	*Dataset* 20 detik
SVM	97%	97%	67%
XGBoost	97%	98%	84%

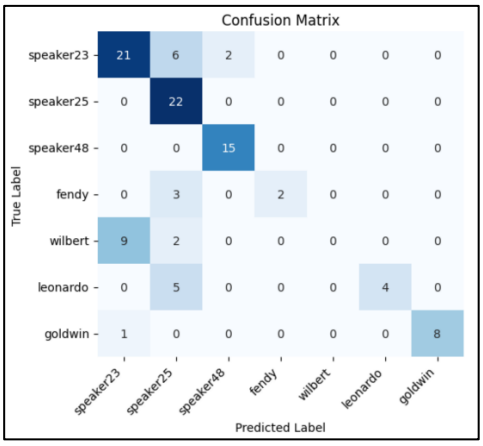
Confusion matrix untuk model SVM ditunjukkan sebagai berikut.



2 seconds

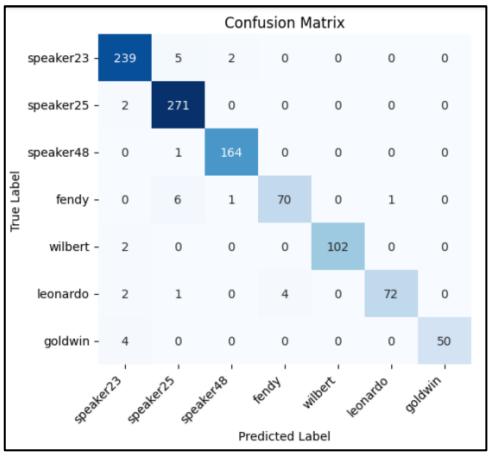


5 seconds

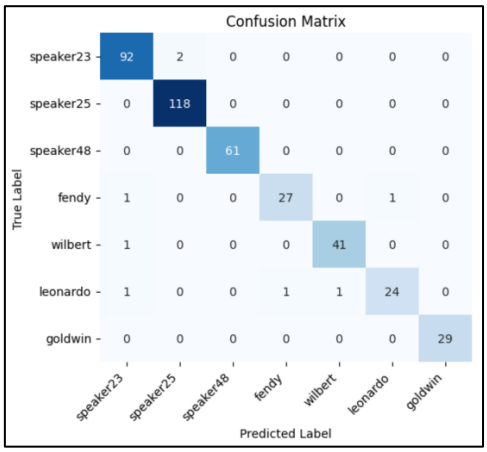


20 seconds

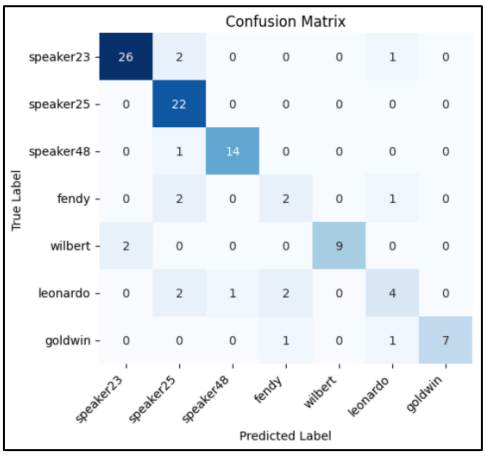
Confusion matrix untuk model XGBoost ditunjukkan sebagai berikut.



2 seconds



5 seconds



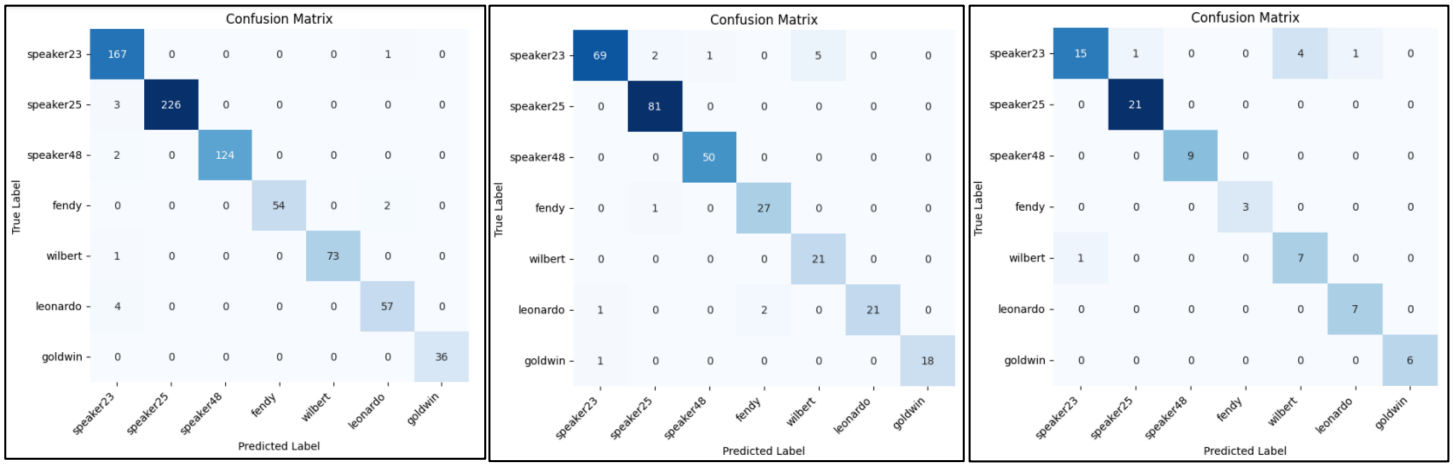
20 seconds

Dari perbandingan yang dilakukan terhadap kedua model berbasis *machine learning* tersebut, terlihat bahwa *dataset* dengan potongan audio sebesar 2 detik selalu menghasilkan nilai akurasi terbesar. Jika dibandingkan, model XGBoost adalah model terbaik yang menghasilkan akurasi mencapai 98% pada *dataset* 2 detik. Nilai akurasi ini terpaut 1% dengan SVM yang hanya mencapai 97% pada *dataset* yang sama. Umumnya, semakin panjang durasi audio dalam suatu *dataset*, maka semakin bagus pula akurasi yang akan didapatkan, karena terdapat lebih banyak data yang dapat dipelajari oleh model dalam setiap file suara. Namun, pada percobaan yang kami lakukan, *dataset* dengan durasi setiap audio selama 20 detik mempunyai akurasi yang lebih rendah dibandingkan dengan *dataset* dengan durasi setiap audio selama 2 detik. Hal ini dikarenakan oleh durasi audio yang terbilang cukup singkat. Maka dari itu, ketika dibagi menjadi *dataset* yang durasinya 20 detik,

jumlah data menjadi sangat sedikit, bahkan 10 kali lebih sedikit dibandingkan dengan *dataset* dengan durasi audio 2 detik. Karena itu, model kurang dapat mempelajari pola pada setiap audio yang ada, sehingga menyebabkan akurasi model lebih rendah pada dataset dengan durasi audio 20 detik.

Model RNN	*Dataset* 2 detik	*Dataset* 5 detik	*Dataset* 20 detik
Training accuracy	98,28%	96,88%	86,74%
Validation accuracy	99%	97,46%	93,24%
Testing accuracy	95,66%	98,26%	90,66%
Testing F1-*score*	98,28%	95,67%	90,62%
Jumlah *epochs*	13	6	11

Confusion matrix untuk model RNN ditunjukkan sebagai berikut.



2 seconds 5 seconds 20 seconds

Selanjutnya untuk model RNN, dari hasil *training* yang dilakukan, terdapat tren yang cukup jelas, yaitu akurasi model pada *dataset* berdurasi 20 detik pasti lebih rendah daripada *dataset* berdurasi 5 detik atau 2 detik. Hal ini kemungkinan besar terjadi karena kecilnya *dataset* yang ada, dengan data setiap *speaker* tidak menyentuh 1 jam, sehingga ketika dibagi menjadi 20 detik, jumlah data yang ada menjadi sangat sedikit dan model akan kesulitan dalam membaca pola-pola data yang ada. Akan tetapi, *dataset* 2 detik dan 5 detik masih menunjukkan performa yang sangat baik dan signifikan, dengan nilai yang tidak berbeda jauh dengan akurasi pada model *machine learning*. Untuk model berbasis RNN, nilai terbaik berada pada *dataset* 5 detik, dengan nilai akurasi *testing* yang lebih besar 0,26% dari akurasi terbaik pada model *machine learning* yang dicapai oleh XGBoost sebesar 98%.

5. KESIMPULAN DAN SARAN DARI HASIL EVALUASI

Secara garis besar, *speaker recognition* dapat diimplementasikan pada sebuah model dengan membagi audio menjadi potongan-potongan kecil berdurasi pendek yang akan diekstraksi menjadi fitur-fitur MFCC. Durasi inilah yang perlu disesuaikan dengan model yang digunakan agar memperoleh akurasi maksimal. Berdasarkan hasil eksperimen, kombinasi model dan durasi sampel audio yang optimal adalah RNN dengan durasi sampel 5 detik, yang menghasilkan akurasi 98,26%. Akan tetapi, jika patokan awalnya adalah *machine learning*, maka kombinasi yang optimal adalah XGBoost dengan durasi sampel 2 detik, yang akurasinya tidak jauh berbeda, yaitu 98%.

Sepanjang hasil eksperimen yang telah dilaksanakan pada *project* ini, model yang dibangun telah menunjukkan performa yang sangat baik dan mengarah pada satu simpulan utama yang jelas. Akan tetapi, untuk membangun sebuah sistem yang ditujukan bagi pendeteksi kehadiran mahasiswa, diperlukan suatu sistem yang mampu mendeteksi audio secara *real-time*. Keluaran dari *project* ini hanya berupa sistem *artificial intelligence* yang sudah mampu memprediksi *speaker* yang sedang berbicara, belum berupa pendeteksi *real-time* secara langsung untuk mengabsen mahasiswa. Sistem ini masih perlu dikembangkan lebih lanjut secara implementatif pada suatu aplikasi agar dapat dimanfaatkan oleh institusi seperti perguruan tinggi. Harapannya, eksperimen dan *project* atau penelitian lain di masa yang akan datang mampu meningkatkan kematangan sistem agar dapat dimanfaatkan secara langsung sebagai sebuah pendeteksi kehadiran mahasiswa.