



**SISTEM KONTROL KECEPATAN KIPAS ANGIN MENGGUNAKAN SUARA  
MULTI – BAHASA MELALUI KLASIFIKASI AUDIO PADA YAMNET**

**TUGAS AKHIR**



**Program Studi**

**S1 Teknik Komputer**

**Oleh:**

**Giga Razki Arianda**

**20410200022**

UNIVERSITAS  
**Dinamika**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2024**

**SISTEM KONTROL KECEPATAN KIPAS ANGIN MENGGUNAKAN SUARA  
MULTI – BAHASA MELALUI KLASIFIKASI AUDIO PADA YAMNET**

**TUGAS AKHIR**



Diajukan Sebagai salah satu syarat untuk menyelesaikan program sarjana

UNIVERSITAS  
**Dinamika**

Oleh:

Nama : Giga Razki Arianda  
NIM : 20410200022  
Program : S1 (Strata Satu)  
Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2024**

**TUGAS AKHIR**  
**SISTEM KONTROL KECEPATAN KIPAS ANGIN MENGGUNAKAN SUARA**  
**MULTI – BAHASA MELALUI KLASIFIKASI AUDIO PADA YAMNET**

Dipersiapkan dan disusun oleh

**Giga Razki Arianda**

**NIM: 20410200022**

Telah diperiksa, dibantu, dibahas, dan disetujui oleh Dewan Pembahas

Pada :

**Susunan Dewan Pembahas**

Pembimbing

I. **Heri Pratikno, M.T., MTCNA., MTCRE.**  
NIDN. 0716117302



Digitally signed by Heri Pratikno, M.T.  
DN: cn=Heri Pratikno, M.T.,  
o=Universitas Dinamika, ou=51 Teknik  
Komputer,  
email=heri@dinamika.ac.id, c=ID  
Date: 2024.03.01 13:31:50 +07'00'  
Adobe Acrobat version: 11.0.23

II. **Musayyannah, S.ST., M.T.**  
NIDN. 0730069102



Digitally signed by Musayyannah  
DN: cn=Musayyannah,  
o=Universitas Dinamika, ou=51  
Teknik Komputer,  
email=musayyannah@dinamika.ac.id,  
c=ID  
Date: 2024.03.01 13:38:39  
+07'00'

Pembahas

I. **Pauladie Susanto, S.Kom., M.T.**  
NIDN. 0729047501



cn=Pauladie Susanto,  
o=Universitas Dinamika, ou=PS 51  
Teknik Komputer,  
email=pauladie@dinamika.ac.id,  
c=ID  
2024.03.01 13:43:34 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar sarjana



Digitally signed by Anjik  
Sukmaaji

Date: 2024.03.05  
11:15:05 +07'00'

**Dr. Anjik Sukmaaji, S.Kom., M.Eng.**

NIDN. 0731057301

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



“GOD IS ALWAYS THERE FOR YOU, NEVER DOUBT”

UNIVERSITAS

- Atta Halilintar

Dinamika



UNIVERSITAS  
Dinamika

Dipersembahkan kepada diri sendiri yang telah berusaha dan tidak menyerah sampai akhir, serta kepada keluarga yang telah mendukung dan mendoakan. Teruntuk manusia – manusia di planet bumi ini yang telah bersedia membantu, mendukung, dan

memotivasi agar dapat menyelesaikan Tugas Akhir ini.

**PERNYATAAN  
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : **Giga Razki Arianda**

NIM : **20410200022**

Program Studi : **S1 Teknik Komputer**

Fakultas : **Fakultas Teknologi dan Informatika**

Jenis Karya : **Laporan Tugas Akhir**

Judul Karya : **SISTEM KONTROL KECEPATAN KIPAS ANGIN MENGGUNAKAN  
SUARA MULTI – BAHASA MELALUI KLASIFIKASI AUDIO PADA  
YAMNET**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non Exclusive Royalty Free Right*) atas seluruh isi / Sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (database) untuk selanjutnya didistribusikan dan dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik hak cipta.
2. Karya tersebut diatas adalah karya asli saya, bukan plagiat baik Sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat Tindakan plagiat pada karya ilmiah ini, saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikina surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 1 Januari 2024



**Giga Razki Arianda**

NIM : 20410200022

## ABSTRAK

Era globalisasi saat ini, terjadi kemajuan teknologi yang sangat pesat dengan ditandainya hampir semua bidang pekerjaan manusia dapat dilakukan secara cepat dan otomatis, dalam rangka mendukung dan memudahkan manusia pada hampir semua bidang pekerjaannya. Sebagai salah satu contoh bagaimana perubahan cara mengoperasikan kipas angin, secara konvensional untuk menghidupkan atau mematikan kipas angin dilakukan dengan cara menekan tombol – tombol yang ada di kipas angin secara manual. Tujuan dari penelitian ini adalah merancang dan membangun sebuah sistem yang dapat mengontrol kecepatan kipas angin menggunakan perintah berbasis suara multi – bahasa, yaitu Bahasa Indonesia, Bahasa Inggris, dan Bahasa Jawa secara *realtime* melalui proses klasifikasi audio pada YAMNet. Hasil penelitian pada proses *training* dengan 12 *epoch* mempunyai akurasi 89%, *loss* 0.71, *ROC curve class* 0 (kipas angin mati) sebesar 97%, *class* 1 (kipas angin putaran rendah) sebesar 100%, *class* 2 (kipas angin putaran sedang) sebesar 95%, dan *class* 3 (kipas angin putaran tinggi) sebesar 99%. Adapun pada proses *training* dengan 15 *epoch* mempunyai akurasi 100%, *loss* sebesar 0.46, *ROC curve class* 0 (kipas angin mati) sebesar 100%, *class* 1 (kipas angin putaran rendah) sebesar 100%, *class* 2 (kipas angin putaran sedang) sebesar 99%, dan *class* 3 (kipas angin putaran tinggi) sebesar 100%. Proses *testing* dari audio dataset *subset test* yang dilakukan sebanyak 40 percobaan menghasilkan akurasi sebesar 90%. Hasil pengujian *testing* menggunakan 12 *epoch* dengan perintah secara langsung dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 10% dan akurasi salah 90%, kipas angin putaran rendah (1) mempunyai akurasi benar 30% dan akurasi salah 70%, kipas angin putaran sedang (2) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, sedangkan kipas putaran tinggi (3) mempunyai akurasi benar 20% dan akurasi salah 80%. Proses *testing* menggunakan 15 *epoch* dengan perintah secara *live* (langsung) dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 53,3% dan akurasi salah 46%. Kipas angin putaran rendah (1) mempunyai akurasi benar 43,3% dan akurasi salah 56,7%, kipas angin putaran sedang (2) mempunyai akurasi benar 36,7% dan akurasi salah 63,3%, sedangkan kipas angin

putaran tinggi (3) mempunyai akurasi benar 43,7% dan akurasi salah 56,7%. proses testing menggunakan 15 *epoch early stopping* dengan perintah secara langsung dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, kipas angin putaran rendah (1) mempunyai akurasi benar 66,7% dan akurasi salah 33,3%, kipas angin putaran sedang (2) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, kipas angin putaran tinggi (3) mempunyai akurasi benar 50% dan akurasi 50%. Hasil testing model dataset subset test menggunakan 12 epoch yang dilakukan sebanyak 40 percobaan pada semua perintah menghasilkan akurasi sebesar 92.5%. Hasil 40 percobaan dataset subset test menggunakan 15 epoch pada semua perintah menghasilkan akurasi sebesar 97.5%. dan hasil 40 percobaan model dataset subset test menggunakan 15 *epoch early – stopping* mendapatkan akurasi sebesar 100%.

**Kata kunci** : Kipas angin, *Audio Classification*, YAMNet



UNIVERSITAS  
**Dinamika**



## KATA PENGANTAR

Puji Syukur ke hadirat Allah SWT yang telah memberikan Rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan Tugas Akhir yang berjudul “Sistem Kontrol Kecepatan Kipas Angin Menggunakan Suara Multi – Bahasa Melalui Klasifikasi Audio Pada YAMNet”.

Menyelesaikan Tugas Akhir ini tidak lepas dari bantuan dari berbagai pihak yang telah memberikan banyak masukan, nasihat, saran, kritik dan dukungan moral maupun materil kepada penulis. Oleh karena itu penulis menyampaikan rasa terima kasih kepada:

1. Orang tua dan keluarga tercinta yang selalu mendoakan mendukung dan memberikan semangat kepada penulis.
2. Bapak Pauladie Susanto.S.Kom.,M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika dan juga selaku Dosen Pembahas yang telah memberikan saran dan izin dalam menyusun Tugas Akhir.
3. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing I yang telah memberikan waktu, ilmu, dan arahan dalam proses Tugas Akhir.
4. Ibu Musayyanah, S.ST., M.T., selaku Dosen Pembimbing II yang telah memberikan waktu, ilmu, dan arahan dalam proses Tugas Akhir.
5. Anselmus Roman, Adisaputra Zidha Noorizki, Rendy Yusuf Prasetyo, Nandang Prayogi, dan Sahrul Rafi Zulfitra selaku sahabat yang selalu memberikan semangat dan bantuan kepada penulis sehingga penulis mampu menyelesaikan Tugas Akhir.
6. Adelia Nuroktaviany Wisan Putri, Sekar Gading Hermawan, Dewi Nabiilah I’zaaz Rahman, Nazila Masyrifani, Faisal Maulana Akbar, Agung Nurullah, Rizal Oktavianto, Claudia Yunita Auliani R, dan 10 orang lainnya yang telah membantu penulis dalam pembuatan audio dataset pada Tugas Akhir.
7. Seluruh teman Angkatan 2020 Teknik Komputer yang telah mendukung penulis dari awal Pendidikan hingga terselesaikan Tugas Akhir.
8. Azizi Shafaa Asadel, Freyanashifa Jayawardana, Gita Sekar Andarini, Reva Fidela Adel Pantjoro, Muhammad Attamimi Halilintar, Jerome Polin Sijabat, dan idol lainnya yang selalu menghibur dan memotivasi.

9. Pihak – pihak lain yang tidak dapat disebutkan satu – persatu yang telah memberikan bantuan dan dukungan kepada penulis.

Semoga Allah SWT memberikan balasan yang setimpal kepada semua pihak yang telah membantu, memberikan bimbingan serta nasehat dalam proses menyelesaikan Tugas Akhir ini. Semoga Tugas Akhir ini dapat diterima dan bermanfaat bagi penulis dan semua pihak.

Surabaya, 1 Maret 2024

Penulis



UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

Halaman

<b>ABSTRAK.....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL.....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
<b>1.1 Latar Belakang.....</b>	<b>1</b>
<b>1.2 Rumusan Masalah.....</b>	<b>3</b>
<b>1.3 Batasan Masalah.....</b>	<b>3</b>
<b>1.4 Tujuan Penelitian.....</b>	<b>4</b>
<b>1.5 Manfaat Penelitian.....</b>	<b>4</b>
<b>Bab II LANDASAN TEORI.....</b>	<b>5</b>
<b>2.1 Kipas Angin.....</b>	<b>5</b>
<b>2.2 Suara.....</b>	<b>6</b>
<b>2.3 YAMNet.....</b>	<b>6</b>
<b>2.4 Data Pre – Processing.....</b>	<b>7</b>
<b>2.5 Python.....</b>	<b>7</b>
<b>2.6 Arduino Uno.....</b>	<b>8</b>
<b>2.7 Modul Relay.....</b>	<b>8</b>
<b>2.8 Arduino IDE.....</b>	<b>9</b>
<b>2.9 Jupyter Notebook.....</b>	<b>10</b>
<b>2.10 Visual Studio Code.....</b>	<b>10</b>
<b>2.11 Waveform.....</b>	<b>11</b>
<b>2.12 Power Supply.....</b>	<b>12</b>
<b>2.13 Kapasitor.....</b>	<b>13</b>
<b>2.14 Long Short Term Memory Bidirectional.....</b>	<b>14</b>
<b>2.15 One – Hot Encoding.....</b>	<b>14</b>
<b>2.16 Menghitung Kinerja.....</b>	<b>15</b>

2.16.1 Akurasi.....	16
2.16.2 Presisi.....	17
2.16.3 F1 -Score.....	17
2.17 Folds.....	18
<b>Bab III METODOLOGI PENELITIAN .....</b>	<b>19</b>
3.1 Perancangan Perangkat Keras .....	19
3.2 Instalasi Environment .....	27
3.3 Data Pre – Paration.....	29
3.3.1 Flowchart Dataset.....	29
3.3.2 Pengumpulan Dataset.....	30
3.3.3 Augmentasi Data.....	31
3.3.4 Karakteristik Audio.....	32
3.3.5 Rename File Dataset.....	33
3.3.6 Pengelompokan Data.....	35
3.3.7 Distribusi Data Kategori.....	37
3.3.8 Datasheet.....	39
3.3.9 Distribusi <i>Folds</i> Subset.....	40
3.3.10 Label Encoding.....	42
3.3.11 Label One – Hot Encoding.....	43
3.4 Data Training.....	45
3.5 Model YAMNet.....	45
3.6 Flowchart .....	48
<b>Bab IV HASIL DAN PEMBAHASAN.....</b>	<b>50</b>
4.1 Hasil Proses Training Model Pertama.....	50
4.2 Hasil Proses Training Model Kedua.....	61
4.3 Hasil Proses Training Model Ketiga.....	73
<b>Bab V KESIMPULAN.....</b>	<b>91</b>
5.1 Kesimpulan.....	91
5.2 Daftar Pustaka.....	93
<b>Daftar Pustaka.....</b>	<b>94</b>
<b>Lampiran .....</b>	<b>97</b>

## DAFTAR GAMBAR

Gambar 2.1 Kipas Angin.....	5
Gambar 2.2 Python.....	7
Gambar 2.3 Arduino Uno .....	8
Gambar 2.4 Modul Relay.....	9
Gambar 2.5 Arduino IDE .....	10
Gambar 2.6 Jupyter Notebook.....	10
Gambar 2.7 Visual Studio Code.....	11
Gambar 2.8 Rangkaian Power Supply.....	12
Gambar 2.9 Arsitektur BiLSTM.....	14
Gambar 2.10 Confusion Matrix.....	16
Gambar 2.11 Folds.....	18
Gambar 3.1 Blok Diagram.....	19
Gambar 3.2 Perancangan Hardware .....	21
Gambar 3.3 Foto Alat.....	22
Gambar 3.4 Rangkaian Skematik.....	26
Gambar 3.5 Load Model YAMNet .....	18
Gambar 3.6 Flowchart audio dataset.....	29
Gambar 3.7 Pengumpulan Dataset.....	31
Gambar 3.8 Augmentasi Data.....	32
Gambar 3.9 Waveform dan Spectogram.....	33
Gambar 3.10 Rename File Audio Dataset .....	34
Gambar 3.11 Pengelompokan Data Berdasarkan Kategori .....	37
Gambar 3.12 Pengelompokan Data Berdasarkan Bahasa .....	37
Gambar 3.13 Distribusi Folds Berdasarkan Label .....	38
Gambar 3.14 Distribusi Folds Berdasarkan Bahasa .....	38
Gambar 3.15 Datasheet .....	40

<b>Gambar 3.16 Distribusi Folds Subset .....</b>	<b>41</b>
<b>Gambar 3.17 Tabel Subset Dataset .....</b>	<b>42</b>
<b>Gambar 3.18 Label Encoding .....</b>	<b>43</b>
<b>Gambar 3.19 One – Hot Encoding .....</b>	<b>44</b>
<b>Gambar 3.20 Struktur Model .....</b>	<b>46</b>
<b>Gambar 3.21 Flowchart Python .....</b>	<b>48</b>
<b>Gambar 3.22 Flowchart Arduino Uno .....</b>	<b>49</b>
<b>Gambar 4.1 Akurasi dan Loss Model Pertama.....</b>	<b>51</b>
<b>Gambar 4.2 Evaluasi Akurasi dan Loss Model Pertama.....</b>	<b>51</b>
<b>Gambar 4.3 Confusion Matrix Model Pertama.....</b>	<b>53</b>
<b>Gambar 4.4 ROC Curve Model Pertama.....</b>	<b>54</b>
<b>Gambar 4.5 Classification Report Model Pertama .....</b>	<b>55</b>
<b>Gambar 4.6 Model Performance Metrics Pertama .....</b>	<b>56</b>
<b>Gambar 4.7 Pengujian Kinerja Model Menggunakan Model Pertama.....</b>	<b>57</b>
<b>Gambar 4.8 Akurasi dan Loss Model Kedua .....</b>	<b>62</b>
<b>Gambar 4.9 Evaluasi Akurasi dan Loss Model Kedua .....</b>	<b>63</b>
<b>Gambar 4.10 Confusion Matrix Model Kedua .....</b>	<b>64</b>
<b>Gambar 4.11 ROC Curve Model Kedua.....</b>	<b>65</b>
<b>Gambar 4.12 Classification Report Model Kedua.....</b>	<b>66</b>
<b>Gambar 4.13 Model Performance Metrics Kedua.....</b>	<b>67</b>
<b>Gambar 4.14 Pengujian Kinerja Model Menggunakan Model Kedua.....</b>	<b>69</b>
<b>Gambar 4.15 Akurasi dan Loss Model Ketiga.....</b>	<b>74</b>
<b>Gambar 4.16 Evaluasi Akurasi dan Loss Model Ketiga .....</b>	<b>75</b>
<b>Gambar 4.17 Confusion Matrix Model Ketiga.....</b>	<b>76</b>
<b>Gambar 4.18 ROC Curve Model Ketiga.....</b>	<b>77</b>
<b>Gambar 4.19 Classification Report Model Ketiga.....</b>	<b>78</b>
<b>Gambar 4.20 Model Performance Metrics Model Ketiga.....</b>	<b>79</b>

**Gambar 4.21 Pengujian Kinerja Model Menggunakan Model  
Ketiga.....80**



UNIVERSITAS  
**Dinamika**

## DAFTAR TABEL

<b>Tabel 4.1</b>	Pengujian Dataset Subset Test Menggunakan Model Pertama.	58
<b>Tabel 4.2</b>	Pengujian Pengguna Suara Yang Terdapat Dalam Model Dataset Pertama.....	59
<b>Tabel 4.3</b>	Pengujian Suara Pengguna Yang Tidak Terdapat Dalam Model Dataset Pertama .....	60
<b>Tabel 4.4</b>	Pengujian Dataset Subset Test Menggunakan Model Dataset Kedua .....	70
<b>Tabel 4.5</b>	Pengujian Pengguna Suara Yang Terdapat Dalam Model Dataset Kedua .....	71
<b>Tabel 4.6</b>	Pengujian Suara Pengguna Yang Tidak Terdapat Dalam Model Dataset Kedua .....	72
<b>Tabel 4.7</b>	Pengujian Dataset Subset Test Menggunakan Model Ketiga ..	81
<b>Tabel 4.8</b>	Pengujian Suara Pengguna Yang Terdapat Dalam Dataset Melalui Mikrofon Menggunakan Model Kedua.....	82
<b>Tabel 4.9</b>	Pengujian Klasifikasi Suara Pengguna Yang Tidak Terdapat Dalam Dataset Melalui Mikrofon Menggunakan Model Kedua.....	83
<b>Tabel 4.10</b>	Perbandingan Akurasi Pengguna Yang Terdapat Dalam Dataset Antar Epoch.....	85
<b>Tabel 4.11</b>	Perbandingan Akurasi Pengguna Yang Tidak Terdapat Dalam Dataset Antar Epoch.....	86
<b>Tabel 4.12</b>	Pengukuran Tegangan Kapasitor Setelah Relay.....	89
<b>Tabel 4.12</b>	Pengukuran Tegangan Kapasitor Seri.....	90



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Di zaman saat ini, kemajuan teknologi yang semakin canggih bekerja secara otomatis diciptakan untuk membantu pekerjaan manusia. Kipas angin merupakan teknologi buatan Dr. Schuyler Skaats Wheeler yang merupakan seorang insinyur dan ilmuwan asal Amerika Serikat (Muazis, 2021). Seiring kemajuan teknologi kipas angin berkembang menjadi kebutuhan teknologi di setiap rumah. Adanya kipas angin, udara di ruangan rumah atau bangunan lebih terjaga. Pada penggunaan kipas seperti menghidupkan dan mematikan ataupun seperti mengaktifkan pilihan lainnya untuk mengatur kecepatan biasanya pengguna harus langsung menghampiri kipas untuk menekan tombol yang berada pada kipas.

*Multi* – Bahasa menjadi landasan dalam mendeteksi berbagai bahasa dari suara pengguna. Bahasa merupakan alat komunikasi yang sering digunakan untuk berkomunikasi, bekerja sama dan mengidentifikasi diri dalam suatu Masyarakat. Masyarakat terdiri dari berbagai status sosial dan latar belakang budaya yang berbeda. Perbedaan akan berdampak pada macam – macam penggunaan bahasa oleh Masyarakat (Sherila, 2012). Sehingga dibutuhkan pengumpulan data mencakup suara bahasa. Data bisa berupa audio yang berbunyi suara perintah dalam berbagai bahasa. Karena itu dibutuhkan mengekstraksi fitur – fitur yang relevan dari informasi tersebut. Fitur – fitur ini bisa meliputi kata – kata dan kalimat bahasa dari variasi bahasa lainnya yang dapat membedakan suara bahasa yang berbeda.

Mikrokontroler saat ini berkembang pesat dan semakin dikenal untuk aplikasi sistem control. salah satu modul mikrokontroler yang sering digunakan adalah Arduino. Arduino merupakan sistem komputer dalam sebuah chip. Arduino memiliki memori, prosesor, I/O dan mikrokontroler, dapat dikatakan sebuah komputer versi *mini* dengan perangkat lunak pendukung pemrograman Arduino Uno (Edowai, 2023).

YAMNet merupakan sebuah model klasifikasi suara menggabungkan arsitektur MobileNetV1 yang sudah dilatih sebelum pengumpulan data audio. Arsitektur ini merupakan *Convolutional Neural Network* (CNN) dengan kemampuan mendalam dalam pemisahan fitur, telah dilatih sebelumnya menggunakan data audio. Keunikan YAMNet terletak pada kemampuannya untuk tidak memerlukan tahap ekstraksi fitur sebelumnya dalam memproses data audio, karena model ini sudah dilengkapi dengan lapisan ekstraksi fitur bawaan. Dengan kata lain, model ini dapat langsung menerima data audio dan mengubahnya ke dalam representasi spektrogram sebelum meneruskan ke MobileNet.

Saat ini Arduino telah banyak digunakan dalam sistem *control* sehingga dapat memenuhi keperluan manusia, tetapi masih sedikit penerapan *audio classification* dengan mikrokontroler Arduino Uno pada YAMNet dalam berbagai hal. Salah satu kontribusi dari Tugas Akhir ini adalah menerapkan Arduino Uno sebagai proses *audio classification*.

Sistem kontrol kipas angin dengan deep learning telah mulai diminati. Seperti pada penelitian yang telah mengembangkan kontrol kipas angin lewat gestur jari dengan Mediapipe (Fakhrudin, 2023). Pengembangan Mediapipe dengan *Automatic Feature Selection* telah dikerjakan oleh (Edowai,2023). Sistem kontrol kipas angin dengan Internet of Things telah berbasis Mediapipe dilakukan oleh Fred (Wakerkwa, 2023).

Tujuan dari Tugas Akhir ini adalah merancang dan membangun sebuah sistem yang dapat mengontrol kecepatan kipas angin dengan suara multi – bahasa melalui *audio classification* pada YAMNet. Proses pengontrolan kecepatan kipas angin dilakukan secara otomatis yang berjalan dalam waktu nyata (*realtime*) berdasarkan pengenalan suara melalui *microphone* dengan tiga bahasa, yaitu: Bahasa Indonesia, Bahasa Inggris dan Bahasa Jawa yang telah dilatih pada model YAMNET *Audio Classification*.

## 1.2 Rumusan Masalah

Dari latar belakang tersebut, dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana mengklasifikasi suara pada YAMNET untuk mengontrol *level* kecepatan kipas angin?
2. Bagaimana mengenali bahasa menggunakan tiga Bahasa (Indonesia, Inggris dan Jawa) pada YAMNET untuk mengontrol *level* kecepatan kipas angin ?

## 1.3 Batasan Masalah

Dalam penelitian ini, terdapat batasan masalah pada beberapa hal sebagai berikut:

1. Pada Tugas Akhir ini tidak membahas jenis suara berdasarkan *gender*
2. Menggunakan suara normal, tidak membahas warna suara (serak, parau dan sebagainya)
3. Tidak membahas intonasi suara
4. Jarak antara pengguna dan mic laptop antara 30 – 40 cm
5. Hanya menggunakan kalimat “kipas mati”, “kipas pejah”, “kipas off”, “kipas tombol satu”, “kipas tombol siji”, “kipas tombol one”, “kipas tombol dua”. “kipas tombol loro”. “kipas tombol two”, “kipas tombol tiga”. “kipas tombol telu”. “kipas tombol three”.

## 1.4 Tujuan

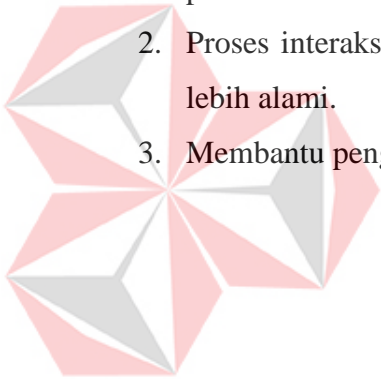
Berdasarkan latar belakang dan rumusan masalah diatas, yang dapat menjadi tujuan pada Tugas Akhir ini sebagai berikut:

1. Mengklasifikasi suara pada YAMNet untuk mengontrol *level* kecepatan kipas angin
2. Mengenali bahasa menggunakan tiga Bahasa (Bahasa Indonesia, Bahasa Jawa dan Bahasa Inggris) pada YAMNet untuk mengontrol *level* kecepatan kipas angin.

## 1.5 Manfaat

Manfaat dari penelitian ini dapat diperoleh sebagai berikut:

1. Memudahkan pengguna dalam mengontrol kecepatan kipas angin hanya dengan perintah suara melalui beberapa bahasa, misalnya pengguna disabilitas.
2. Proses interaksi antara pengguna dan kipas angin dapat dilakukan secara natural dan lebih alami.
3. Membantu pengguna disabilitas yang tidak mempunyai tangan



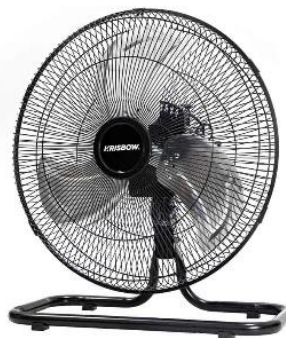
UNIVERSITAS  
Dinamika

## BAB II

### LANDASAN TEORI

#### 2.1 Kipas Angin

Kipas angin merupakan alat penghasil angin yang digunakan untuk Pendingin udara, penyegar udara, ventilasi dan pengering. Kipas angin memiliki dua jenis yaitu kipas angin tradisional seperti kipas angin tangan dan kipas angin listrik seperti angin yang digerakkan menggunakan tenaga listrik. Seiring berjalannya waktu perkembangan kipas semakin memiliki variasi dari segi ukuran, penempatan posisi dan fungsi. Dari segi ukuran kipas angin memiliki beberapa variasi dari kipas angin mini yang dipegang tangan menggunakan baterai, kipas yang digunakan pada dalam CPU komputer seperti kipas angin untuk mendinginkan komponen – komponen komputer seperti prosesor, kartu grafis dan lain – lainnya berfungsi untuk menjaga suhu udara agar tidak melewati batas suhu yang ditetapkan. Kipas angin memiliki 3 cara untuk mengontrol kecepatan kipas angin yaitu menggunakan pemutar, tali penarik dan *remote control*. Kipas angin juga memiliki dua perputaran baling – baling diantaranya *centrifugal* yang mengalirkan angin searah dengan poros kipas dan *axial* yang mengalirkan angin secara paralel dengan poros kipas, (Lestari, et al., 2020).



Gambar 2.1 Kipas Angin

## 2.2 Suara

Karena kebanyakan orang menggunakan komputer untuk berkomunikasi, suara adalah salah satu bentuk komunikasi yang paling efektif dan paling banyak digunakan. Telinga manusia dapat mendengar suara dengan frekuensi antara 20 Hz dan 20 KHz. Frekuensi paling rendah adalah 20 Hz, dan frekuensi paling tinggi adalah 20 KHz. Tiga komponen utama membentuk gelombang suara yaitu frekuensi, amplitudo, dan periode (Armaputra, 2022).

## 2.3 *Yet Another Multi Layer Network (YAMNet)*

YAMNet merupakan singkatan dari “*Yet Another Multi Layer Network*” mewakili sebuah model klasifikasi suara yang mengintegrasikan arsitektur MobileNetV1. Arsitektur ini merupakan *Convolutional Neural Network (CNN)* dengan kemampuan mendalam dalam pemisahan fitur, telah dilatih sebelumnya menggunakan data audio. Keunikan YAMNet terletak pada kemampuannya untuk tidak memerlukan tahap ekstraksi fitur sebelumnya memproses data audio, karena model ini sudah dilengkapi dengan lapisan ekstraksi fitur bawaan. Dengan kata lain, model ini dapat langsung menerima data audio dan mengubahnya ke dalam representasi spektrogram sebelum meneruskan ke MobileNet.

Arsitektur MobileNetV1 pada YAMNet memungkinkan konversi data audio ke dalam bentuk spektrogram melalui ekstraksi fitur bawaan tersebut. Spektrogram ini kemudian diteruskan ke dalam MobileNet untuk proses klasifikasi lebih lanjut. Perlu dicatat bahwa salah satu persiapan data yang diperlukan oleh YAMNet adalah melakukan pengambilan sample ulang terhadap klip audio, mengubah frekuensi sampelnya menjadi 16000 Hz, dan memastikan bahwa audio tersebut hanya memiliki satu saluran (*audio channel tunggal*). Langkah ini diambil sebagai bagian dari proses persiapan data untuk meningkatkan akurasi dan konsistensi performa model.

(Malmberg, 2021) telah menggarisbawahi pentingnya persiapan data ini dalam konteks YAMNet. Pengambilan sample ulang klip audio menjadi 16000 Hz dan penggunaan saluran tunggal tidak hanya mempersiapkan data agar sesuai dengan input yang diharapkan oleh model, tetapi juga meningkatkan efektivitas model dalam melakukan klasifikasi suara.

Dengan demikian, proses tersebut menjadi langkah kritis dalam menjamin keberhasilan dan ketepatan klasifikasi audio yang dilakukan oleh YAMNet.

#### 2.4 Data Pre – processing

Audio dapat digunakan untuk merekam audio, dan data dapat digunakan untuk melatih model dalam periode waktu yang lebih panjang, memungkinkan model untuk mengumpulkan informasi yang lebih banyak dan mendalam dari dataset yang tersedia. Cara umum melakukan ini adalah dengan cara menerapkan *Short – Time Fourier Transform* (STFT) pada sinyal suara dan mengubah hasil yang berasal dari STFT menjadi spektrogram atau spektrogram Mel yang menerapkan skala logaritmik. Pada struktur YAMNet *Audio Classification* menggunakan model YAMNet, maka YAMNet dapat dipakai untuk menjalankan data *pre – processing*. Dengan mengambil sampel ulang rekaman suara menjadi saluran suara tunggal 16000 Hz yang dapat dimasukkan ke dalam model YAMNet yang dapat menghasilkan spektrogram Mel, spektrogram tersebut lalu diteruskan ke mobilenet untuk menghasilkan *embeddings* yang dapat digunakan untuk pelatihan model baru.

#### 2.5 Python

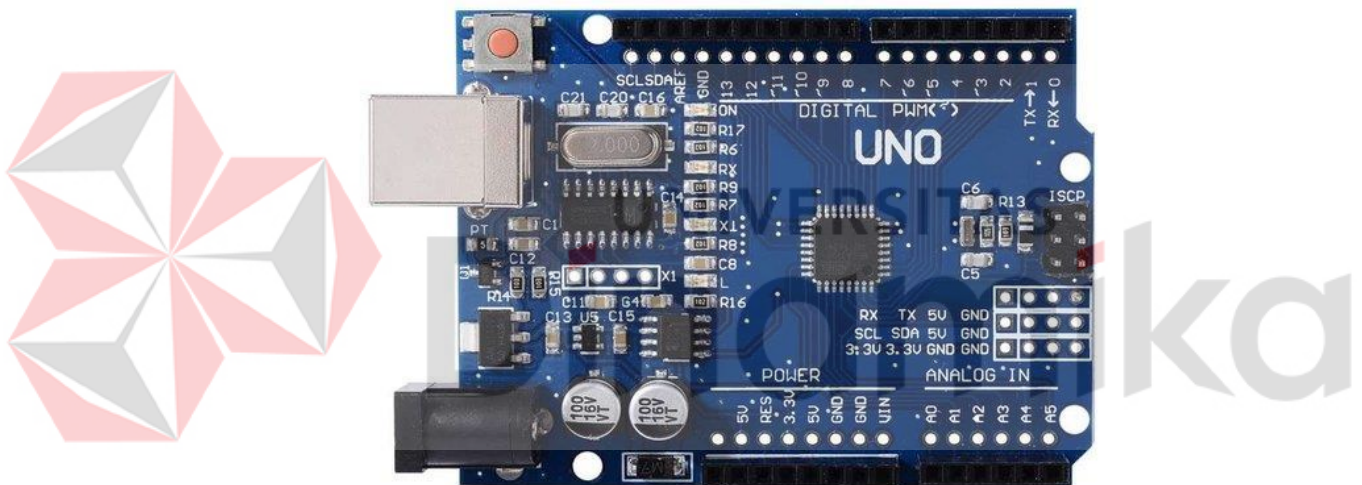
Python merupakan bahasa pemrograman *interpretative* yang mampu menerapkan berbagai kebutuhan pemrograman dengan filosofi rencana fokus pada tingkat keterbacaan kode dan salah satu bahasa yang digunakan untuk pengembangan AI, ilmu informasi dan *Internet of Things*. Dalam contoh ini, bahasa pemrograman python digunakan untuk membuat pelatihan *dataset* dan proses klasifikasi suara menggunakan tiga bahasa.



Gambar 2.2 Python

## 2.6 Arduino Uno

Arduino Uno merupakan mikrokontroler berbasis ATmega328P. Mikrokontroler tersebut memiliki 14 pin *input/output advanced* (terdapat 6 pin yang dapat digunakan sebagai *input* PWM). 6 input analog, kristal kuarsa 16Hz, konektor daya, konektor USB, *header* ICSB dan tombol *reset*. Pada Arduino Uno berisi semua yang diperlukan untuk mendukung mikrokontroler, cukup sambung ke komputer dengan kabel USB sebagai sumber daya dengan adaptor AC – DC.



Gambar 2.3 Arduino Uno

## 2.7 Modul Relay

Relay merupakan komponen elektromekanikal yang memiliki bagian utama yakni *electromagnet (coil)* dan mekanikal (seperangkat kotak saklar), transfer menerapkan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik yang bertegangan tinggi. Contohnya, transfer yang menggunakan *electromagnet* 5 VDC dan 50 mA mampu menggerakkan *Ameture* Relay (berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A (Lubis, 2019).





Gambar 2.4 Modul Relay

## 2.8 Arduino IDE

Arduino *Integrated Development Environment* (IDE) merupakan perangkat lunak yang digunakan untuk menulis program, meng-*compile*, dan mengunggah ke papan Arduino. Arduino terdiri dari teks editor untuk penulisan kode, area pesan, console teks *toolbar* dengan tombol – tombol untuk fungsi umum dan sederetan menu. Perangkat lunak yang di program menggunakan Arduino dinamakan outlines. *Sketches* ditulis di teks editor dan disimpan dengan file berekstensi *ino*. Editor teks ini memiliki fitur untuk *cut/paste* dan *search/replace*. Area pesan terdapat umpan balik ketika menyimpan dan mengunggah dokumen, dan menunjukkan jika terjadi *error* (Lestari, et al., 2020). Arduino IDE juga terdapat dengan *library C/C++* yang disebut *wiring* untuk membuat operasi input dan *yield* menjadi lebih mudah. Perkembangan Arduino IDE bermula dari *software processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman Arduino.



Gambar 2.5 Arduino IDE

## 2.9 Jupyter Notebook

Jupyter *Notebook* adalah alat terkenal untuk bekerja dengan data menggunakan bahasa pemrograman Python. Jupyter *Notebook* memungkinkan untuk mengintegrasikan antara kode dengan hasil dalam satu dokumen secara interaktif, Jupyter adalah organisasi *non – profit* yang mengembangkan perangkat lunak interaktif dalam berbagai jenis bahasa pemrograman. Sedangkan *notebook* merupakan satu perangkat lunak buatan Jupyter, berupa aplikasi *web* sumber terbuka yang memungkinkan pengguna untuk membuat dan berbagi dokumen interaktif berisi *live code*, persamaan, visualisasi, dan teks naratif yang kaya (Wijoyo, 2021).



Gambar 2.6 Jupyter Notebook

## 2.10 Visual Studio Code

*Visual Studio Code* merupakan aplikasi teks editor *open source* yang diciptakan oleh Microsoft yang mampu dijalankan pada sistem operasi Windows, Linux, dan MacOS. Ini adalah jenis kode yang dapat digunakan dengan berbagai bahasa pemrograman, termasuk

C++, C#, Java, Python, PHP, dan GO. Perangkat lunak tersebut memiliki kemampuan mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian kode tersebut. Visual studio code telah terintegrasi ke Github, fitur lainnya terdapat kemampuan untuk menambahkan ekstensi kemampuan untuk menambahkan ekstensi bagi para pengembang yang ingin menambah ekstensi untuk menambahkan fitur yang tidak ada di Visual Studio Code (Ramadhan, et al., 2019).



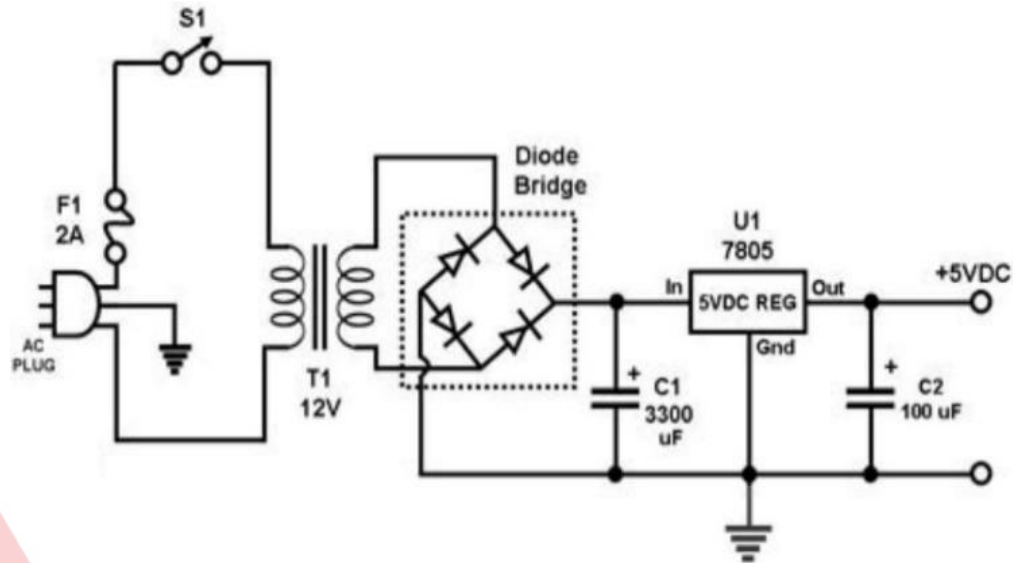
Gambar 2.7 Visual Studio Code

### 2.11 WAV (Waveform)

WAV merupakan jenis format audio yang telah diciptakan oleh Perusahaan Microsoft dan IBM. Format WAV adalah sinyal audio berjenis raw format (kasar) yang direkam dan dikuantisasi menjadi data digital. WAV dipakai sebagai penyimpanan data audio mentah pada windows yang memiliki kapasitas sekitar 10 MB (Uber Byte) per menit dan kapasitas maksimal 2 GB (Giga Byte). Pemicu file WAV memiliki kapasitas yang besar karena tidak ada kompresi file sehingga file jenis tersebut digunakan untuk audio ataupun video (Armaputra, 2022).

## 2.12 Power Supply

Gambar 2.8 merupakan rangkaian sederhana *power supply*.



Gambar 2.8 Rangkaian *Power Supply* (source: Griyasis)

*Power Supply* atau sumber tegangan adalah suatu alat atau sistem yang dapat menghasilkan arus listrik. Berikut ini adalah jenis *power supply*:

### 2.12.1 Sumber Arus Searah (*Direct Current/DC*)

Arus listrik searah adalah arus listrik yang bernilai konstan dan mengalir dari potensial tinggi (+) ke potensial rendah (-). Besar arus listrik searah yang sering ditemukan berkisar antara 1,5 hingga 24 volt. Arus listrik searah biasa digunakan pada baterai, dinamo arus searah, dan aki. Sumber tegangan searah merupakan sumber tegangan yang tidak mengalami perubahan terhadap waktu.

### 2.12.2 Sumber Arus Bolak – Balik (*Alternating Current/AC*)

Arus listrik bolak - balik adalah arus listrik dengan besar dan arah yang berubah-ubah secara bolak-balik. Arus AC mengalir bolak-balik dari potensial tinggi (+) ke potensial rendah (-) dan dari potensial rendah (-) ke potensial tinggi (+). Dalam sedetik, arus AC berbolak-balik sebanyak 50 hingga 60 kali. Gelombang listrik pada arus bolak balik berbentuk sinusoidal, gelombang segi empat, atau gelombang segitiga. Contoh penggunaan

arus listrik bolak-balik yaitu pada jaringan PLN dan generator AC. Arus listrik bolak-balik berkisar antara 110 hingga 220 volt dengan frekuensi 50 Hz.

Penggunaan arus bolak-balik (AC) pada rangkaian tidak dilakukan secara langsung, tetapi harus diubah terlebih dahulu menjadi arus searah (DC). Alat yang digunakan untuk mengubah tegangan listrik bolak-balik menjadi tegangan listrik searah dinamakan adaptor. Adaptor dapat mengeluarkan tegangan searah dengan nilai yang berbeda – beda, mulai dari 1,5 hingga 12 volt, dan dapat diperbesar sesuai dengan kebutuhan.

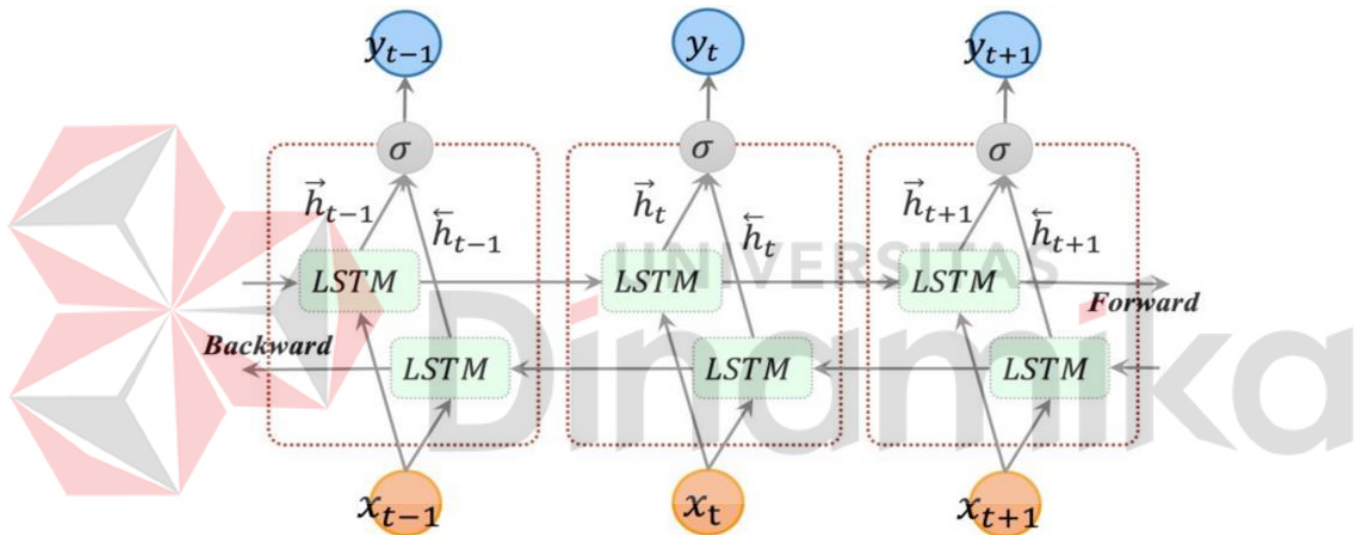
### 2.13 Kapasitor

Kapasitor (Kondensator) yang dalam rangkaian elektronika dilambangkan dengan huruf “C”, merupakan suatu alat yang dapat menyimpan energi/muatan listrik di dalam medan listrik, dengan cara mengumpulkan ketidakseimbangan internal dari muata listrik (Ramdhani, 2005). Kapasitor ditemukan oleh Michael Faraday. Dengan satuan Farad (F). Satu Farad =  $9 \times 10^{11} \text{ cm}^2$  yang artinya luas permukaan kepingan tersebut. Kapasitansi sebesar 1 Farad jika dengan tegangan 1 volt dapat memuat muatan *electron* sebanyak 1 Coulomb atau setara dengan  $6,25 \times 10^{18}$  elektron. Kondensator ditandai dengan memiliki dua kaki dan dua kutub, yaitu positif (+) dan negatif (-) serta kapasitor yang digunakan untuk mengontrol listrik. Kapasitor yang digunakan untuk mengontrol kapasitas lebih rendah, namun tidak memiliki kutub yang positif (+) atau negatif (-) pada kebanyakan berbentuk bulat pipih berwarna coklat, merah, hijau seperti tablet.

Farad merupakan satuan dari kapasitor. Sistem kapasitif merupakan sistem yang dapat menyimpan muatan atau medan listrik. Sedangkan kapasitor merupakan sistem kapasitif yang dibuat supaya mempunyai harga kapasitansi tertentu. Kapasitor yang digunakan untuk memperbesar faktor daya dipasang paralel dengan rangkaian beban. Apabila rangkaian tersebut diberi tegangan maka *electron* akan mengalir masuk ke kapasitor. Saat kapasitor penuh dengan muatan *electron* maka tegangan akan berubah. Kemudian *electron* akan ke luar dari kapasitor dan mengalir ke dalam rangkaian yang memerlukannya, dengan demikian pada saat itu kapasitor membangkitkan daya reaktif. Apabila tegangan berubah tersebut kembali normal, kapasitor akan menyimpan *electron* kembali.

## 2.14 Bidirectional Long Short Term Memory (BiLSTM)

*Bidirectional Long Short-Term Memory* memiliki dua jaringan LSTM yang pertama jaringan berfungsi dalam memproses urutan masukan data ke arah depan (*forward*) dan jaringan LSTM kedua berfungsi dalam memproses urutan data dari arah sebaliknya (*backward*). Kemudian output dari jaringan LSTM *forward* dan *backward* digabungkan pada setiap urutan waktu. Dengan adanya dua *layer* yang berlawanan arah tersebut, model dapat mempelajari informasi masa lalu dan informasi masa depan untuk setiap *sequence input* (Puteri, 2023). Arsitektur BiLSTM mengacu pada (Cui et al., 2020) dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arsitektur BiLSTM

Dalam arsitektur BiLSTM terdiri dari dua bagian, yaitu bagian depan (*forward*) dan bagian belakang (*backward*). Bagian depan memproses informasi dari kiri ke kanan, sedangkan bagian belakang memproses informasi dari kanan ke kiri.

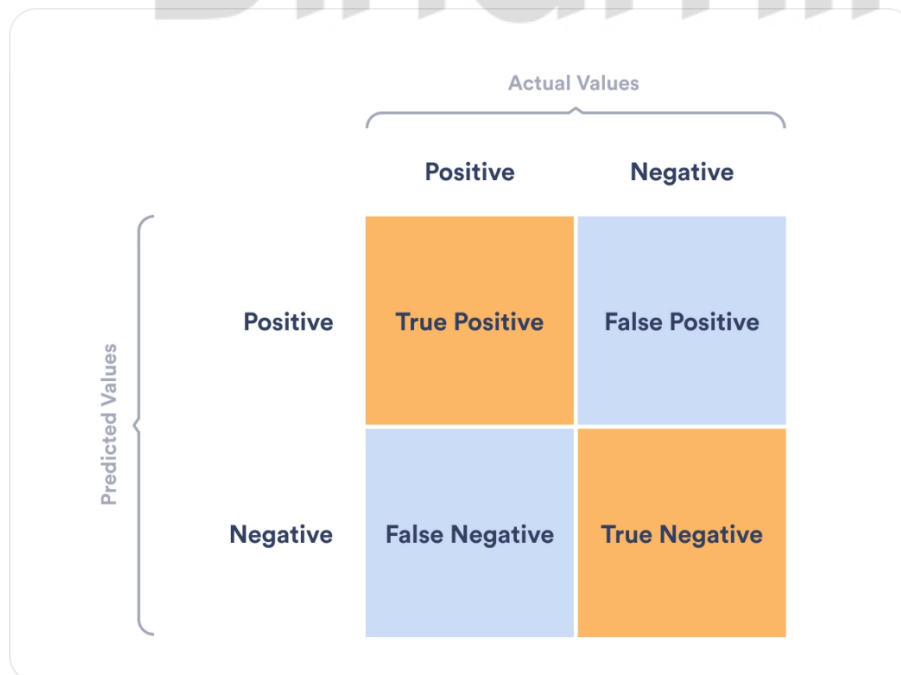
## 2.15 One – Hot Encoding

*One – Hot Encoding* merupakan teknik perubahan kerangka data kategori menjadi kerangka biner, dengan panjang sesuai dengan banyaknya kelompok data yang berbeda

Potdar et al., 2017). Perubahan label menggunakan *One-Hot Encoding* pada umumnya dipakai saat bekerja memakai algoritma jaringan saraf tiruan, karena hasil *output* dari setiap label dihitung secara terpisah. *One-Hot Encoding* dilakukan dengan cara mengganti nilai dari setiap *class* menjadi nilai biner, dengan indeks *class* yang bersangkutan diberi nilai 1 dan yang lain diberi nilai 0.

## 2.16 Confusion Matrix

*Confusion Matrix* adalah alat evaluasi kinerja yang sering digunakan dalam konteks klasifikasi. Matriks tersebut menggambarkan hasil klasifikasi model dengan membandingkan prediksi yang benar (*true positive* dan *true negative*) dengan yang salah (*false positive* dan *false negative*). Dalam konsep ini, terdapat sejumlah Matrix evaluasi performa yang dihasilkan dari *confusion matrix*, seperti *precision*, *recall*, dan *f1 – score*. *Precision* mengukur proporsi dari hasil positif yang benar dari total hasil positif, sementara *recall* mengukur proporsi dari hasil positif yang benar terhadap keseluruhan kelas yang seharusnya positif. *F1 – score*, sebagai gabungan *precision* dan *recall*, memberikan ukuran *holistic* terhadap kinerja model klasifikasi. Parameter ini membantu memberikan pemahaman yang lebih komprehensif mengenai performa model dalam mengatasi tantangan klasifikasi dalam berbagai situasi dan *dataset*.



Gambar 2.10 *Confusion Matrix* (source: plat AI)

### 2.16.1 Akurasi

Akurasi adalah sebuah parameter penting yang berguna untuk mengukur perhitungan performa yang paling intuitif dan hanya rasio pengamatan yang diprediksi dengan benar terhadap total pengamatan. Akurasi merupakan ukuran yang bagus tetapi hanya jika model menggunakan data simetris dimana jumlah data pada setiap *class* adalah hampir sama. Perhitungan pada nilai akurasi bakal menghasilkan nilai antara 0 dan 1, namun ini bisa dipahami juga dengan nilai persentase. Untuk menghitung akurasi menggunakan persamaan (2.1) seperti dibawah ini (Riberu, 2023).

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.1)$$

### 2.16.2 Presisi

Presisi merupakan *matrix* evaluasi performa dalam konteks klasifikasi yang mengukur proporsi dari hasil positif yang benar (*true positives*) dari keseluruhan hasil yang diprediksi sebagai positif (*true positive* dan *false positive*). Dengan kata lain, presisi memberikan informasi tentang sejauh mana model dapat menghindari memberikan label positif pada data yang seharusnya negatif. Presisi dihitung dengan rumus (2.2).

$$Presisi = \frac{True\ Positives}{True\ Positives+False\ Positives} \quad (2.2)$$

Presisi memiliki nilai antara 0 dan 1, dimana nilai lebih tinggi menunjukkan kemampuan model untuk memberikan prediksi positif yang akurat. *Matrix* ini berguna ketika dampak dari *false positives* tinggi atau ketika penting untuk menghindari kesalahan dalam mengklasifikasikan data sebagai positif.



### 2.16.3 *F1 – Score*

*F1 – Score* adalah metrik evaluasi klasifikasi yang menggabungkan presisi dan *recall* untuk memberikan ukuran *holistic* tentang performa model. *Matrix* ini bermanfaat dalam situasi dimana perlu menjaga keseimbangan antara presisi dan *recall*, terutama saat didistribusi kelas tidak seimbang. *F1 – score* dihitung dengan rumus (2.3).

$$F1 - score = \frac{2 \times \text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (2.3)$$

### 2.17 *Folds*

*Folds* adalah teknik *cross – validation* yang digunakan untuk mengevaluasi kinerja model pembelajaran mesin. Pada *folds*, data dibagi menjadi beberapa bagian yang disebut *fold*. Model kemudian dilatih pada beberapa *fold* dan diuji pada yang tersisa. Proses ini diulang beberapa kali, dengan setiap *fold* digunakan sebagai *fold* pengujian sekali. Teknik *folds* memiliki beberapa keunggulan dibanding dengan Teknik *cross-validation* lainnya, yaitu:

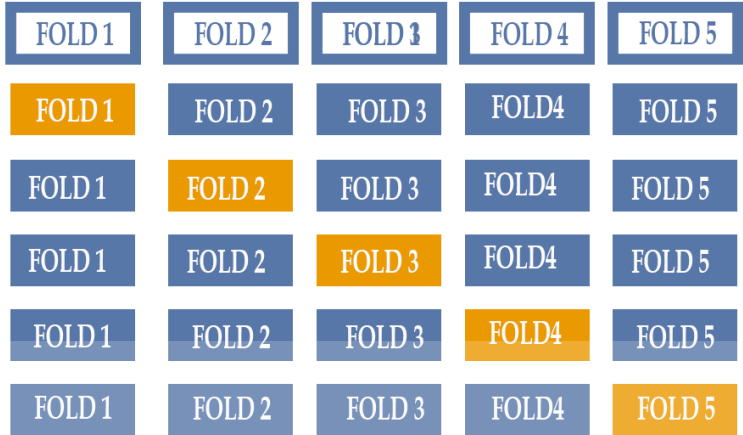
- Lebih akurat dalam memperkirakan kinerja model pada data yang tidak diketahui.
- Lebih stabil, karena tidak bergantung pada urutan data yang digunakan untuk pelatihan.

Pada model YAMNet, *folds* digunakan untuk mengatasi masalah *overfitting*. *Overfitting* adalah masalah yang terjadi ketika model terlalu cocok dengan data pelatihan, sehingga tidak dapat menghasilkan hasil yang baik pada data yang tidak diketahui. Dengan menggunakan *folds*, model YAMNet dapat dilatih data yang lebih besar, sehingga dapat mengurangi resiko *overfitting*.

SEMUA DATA

DATA TRAINING

DATA TEST



UNIVERSITAS  
Dinamika

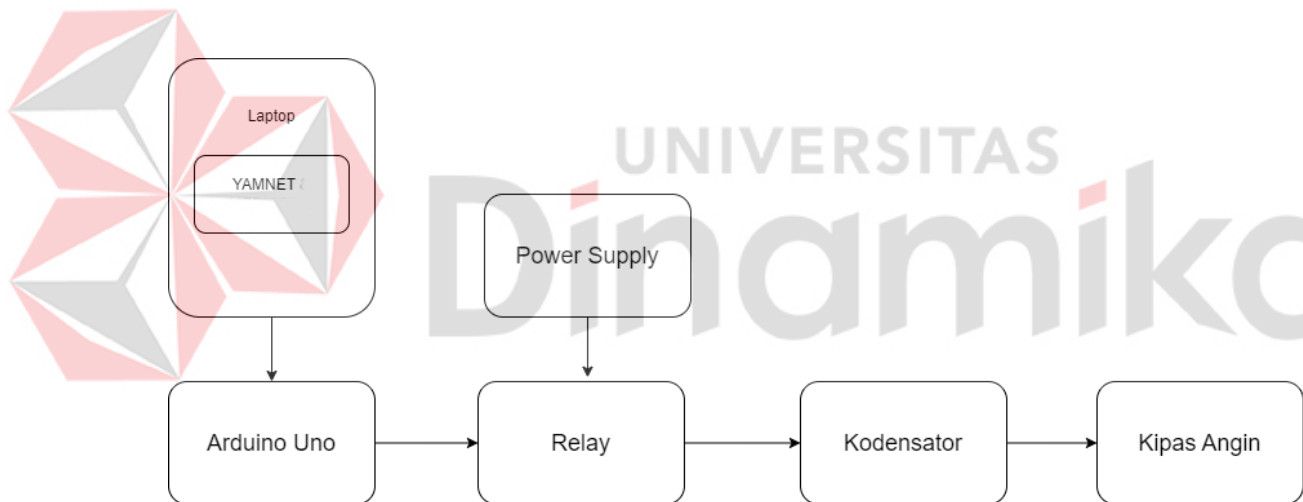
Gambar 2.11 *Fold Dataset*

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Perancangan Perangkat Keras

Pada Gambar 3.1 merupakan struktur perangkat keras, dan *inputnya* yang berasal dari mikrofon laptop untuk mengenali suara pengguna. Setelah itu input tersebut diproses oleh komputer menggunakan sistem *audio classification* dengan model YAMNet kemudian *output* tersebut akan terhubung ke mikrokontroler Arduino Uno untuk melakukan aksi pada relay dengan mematikan ataupun mengatur kecepatan kipas angin dari deteksi suara berdasarkan bahasa yang telah dilatih.

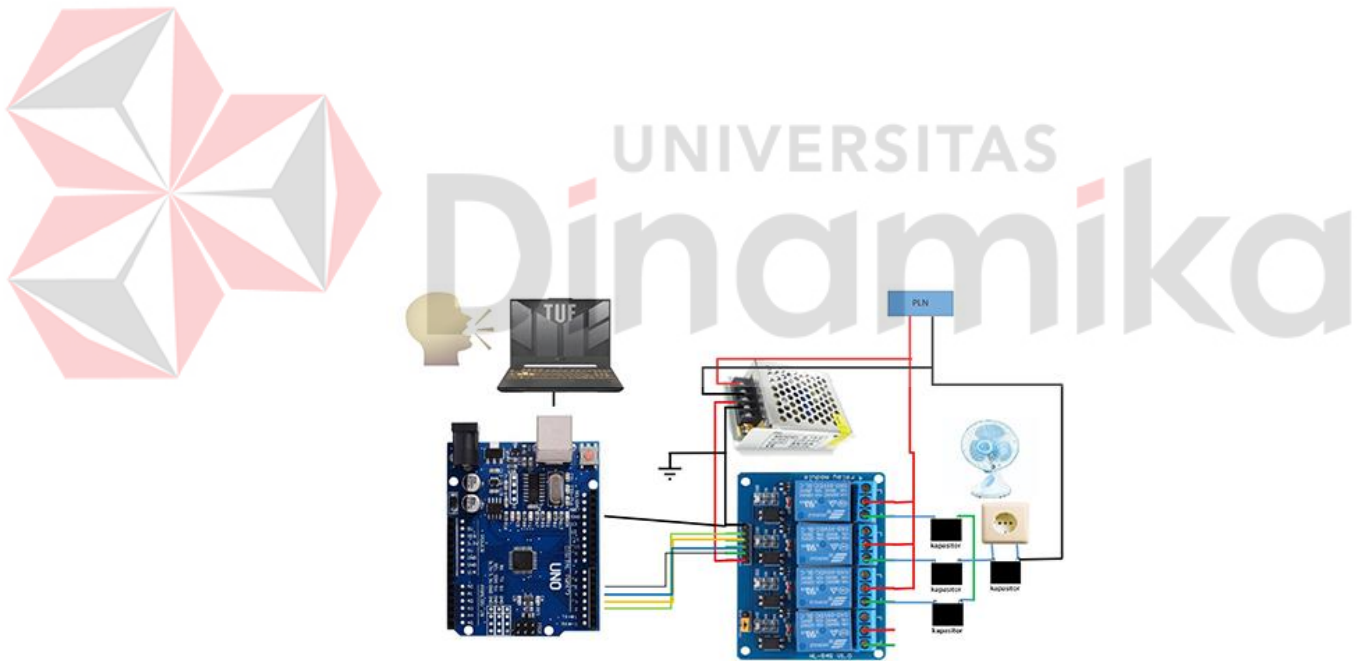


Gambar 3.1 Blok Diagram

Kemudian pada rangkaian skematik mikrokontroler Arduino Uno dihubungkan ke relay. Lalu relay akan dihubungkan dengan kodensator untuk menyimpan daya listrik dan mengatur *level* kecepatan kipas angin, mulai dari nilai 0 ataupun kipas angin tidak nyala, nilai satu atau kecepatan rendah, nilai dua atau kecepatan sedang dan nilai tiga atau kecepatan tinggi. Pada Gambar 3.1 merupakan perancangan perangkat keras yang akan dilakukan penelitian dalam Tugas Akhir ini, cara kerja perangkat keras tersebut pertama-tama pengguna melakukan perintah suara melalui mikrofon komputer kemudian akan diproses oleh YAMNET menggunakan fitur *audio classification* pada komputer yang

memiliki 4 *label* yaitu Mati, Rendah, Sedang dan Tinggi. Jika perintah suara dari pengguna “kipas mati”, “kipas pejah”, “kipas *off*” maka akan terdeteksi mati, jika perintah suara dari pengguna “kipas tombol satu”, “kipas tombol siji”, “kipas tombol *one*” maka akan terdeteksi Rendah, jika perintah suara dari pengguna “kipas tombol dua”, kipas tombol loro”, “kipas tombol *two*” maka akan terdeteksi sedang, jika perintah suara dari pengguna “kipas tombol tiga”, “kipas tombol telu”, “kipas tombol *three*” maka akan terdeteksi tinggi.

NO merupakan singkatan dari *Normally Open*, sedangkan NC merupakan singkatan dari *Normally Close*. Maksudnya adalah ketika kondisi diam/mati/ tidak diberi tegangan, maka pin COM (*Common Contact*) dan NO tidak terhubung(*open*). Sedangkan pin COM dan NC terhubung (*close*). Namun ketika relay hidup atau diberi tegangan maka NO akan berubah menjadi NC, dan NC akan berubah menjadi NO. Inilah yang sering digunakan sebagai pemutus / penghubung arus listrik (saklar).



Gambar 3.2 Perancangan *Hardware*

Gambar 3.2 model perancangan ini merangkai komponen perangkat keras untuk menciptakan suatu sistem yang terdapat memenuhi kebutuhan penelitian. Melalui proses ini, berbagai elemen perangkat keras diintegrasikan secara hati-hati untuk mencapai tujuan

penelitian yang telah ditetapkan. Pada tahap ini, penggabungan komponen-komponen tersebut dibuat sesuai dengan spesifikasi dan desain yang telah disusun sebelumnya.

Proses perancangan *hardware* ini melibatkan pemilihan dan penyesuaian berbagai elemen, seperti sensor, mikrokontroler dan komponen pendukung lainnya. Setiap bagian dari hardware tersebut diatur dan dihubungkan dengan cermat untuk memastikan kerja yang sinergis dan efisien. Tujuan utama dari model perancangan *hardware* ini adalah untuk mendukung pelaksanaan eksperimen dan pengumpulan data dengan akurat dan konsisten.

Dalam konteks ini, desain *hardware* menjadi elemen kunci yang dapat mempengaruhi keberhasilan pengumpulan data dan pencapaian tujuan penelitian. Oleh karena itu, keakuratan dan kehandalan hardware menjadi fokus utama dalam metodologi penelitian ini. Gambar 3.3 menjadi panduan alat fisik yang membantu dalam memahami struktur dan konfigurasi *hardware* yang diterapkan dalam eksperimen.



Gambar 3.3 Foto Rangkaian

Pada Gambar 3.3 merupakan rangkaian relay pada *level* rendah menggunakan kapasitor motor sebesar 1,2  $\mu\text{F}$  / 400 V untuk kecepatan paling rendah kipas angin. Selanjutnya, pada *level* sedang relay menggunakan kapasitor motor 2  $\mu\text{F}$  / 400 V, yang sesuai dengan kecepatan sedang kipas. Sementara itu, pada *level* tinggi menggunakan kapasitor 2  $\mu\text{F}$  / 400 V juga digunakan untuk menunjukkan bahwa kipas beroperasi pada kecepatan tinggi. Ketiga

kapasitor motor ini dihubungkan secara paralel, memungkinkan modulasi tegangan yang variative untuk mengatur kecepatan kipas. Selanjutnya hasil dari hubungan paralel ini dihubungkan secara seri dengan kapasitor motor 1 uF / 400V.

Fungsi utama dari ketiga kapasitor motor yang dihubungkan secara paralel adalah untuk menciptakan nilai kapasitansi total yang sesuai dengan kecepatan yang diinginkan. Kapasitor motor pada level rendah memberikan kontribusi terhadap pembentukan kecepatan rendah kipas, sementara kapasitor pada level sedang dan tinggi memberikan kontribusi masing – masing terhadap kecepatan sedang dan tinggi. Dengan menghubungkan ketiganya secara paralel, peneliti dapat menggabungkan efek dari masing-masing kapasitor motor untuk mencapai kecepatan yang sesuai dengan kebutuhan.

Selanjutnya hasil dari hubungan paralel tersebut dihubungkan secara seri dengan kapasitor motor sebesar 1 uF / 400 V. Fungsi dari kapasitor ini adalah untuk memberikan nilai kapasitansi tambahan yang diperlukan untuk menyesuaikan kecepatan kipas angin sesuai dengan kebutuhan pengguna. Dengan menggabungkan konsep hubungan paralel dan seri pada kapasitor motor, rangkaian ini menciptakan *control* yang efektif untuk mengatur kecepatan kipas angin, memberikan fleksibilitas dan adaptabilitas dalam mengatasi perubahan kecepatan yang diinginkan oleh pengguna. Dengan demikian, rangkaian ini tidak hanya memberikan control kecepatan yang efisien tetapi juga memberikan solusi yang andal dan efektif untuk pengaturan kipas angin berdasarkan *level* kecepatan. Selain itu, ketika tombol fisik kipas angin berada pada *level* rendah, *relay channel* 3 diaktifkan, memungkinkan kapasitor motor sebesar 2 uF / 400 V terlibat dalam modulasi tegangan. Kapasitor tersebut digunakan untuk level kecepatan tinggi, berfungsi sebagai pengatur kecepatan standar untuk *level* rendah sesuai dengan kebutuhan pengguna. Dengan demikian, rangkaian ini memberikan fleksibilitas dan *control* yang optimal terhadap kecepatan kipas angin, mengadaptasikan diri dengan kebutuhan penggunaan dan preferensi tingkat kecepatan yang diinginkan.

Untuk mencari reaktansi kapasitif dari masing – masing kapasitor menggunakan persamaan (3.1).

Rumus Reaktansi Kapasitif =  $X_c$

$$X_c = \frac{1}{2 \times \pi \times f \times C} \quad (3.1)$$

Keterangan :  $3,14 = \Phi$   
 $f =$  Frekuensi listrik 50 Hz  
 $C =$  Besar kapasitas kapasitor dalam Farad  
1 Farad = 1,000,000  $\mu\text{F}$  = satu juta microfarad

$X_{c1}$  atau Kapasitor 1,2  $\mu\text{F}$  :

$$X_{c1} = \frac{1,000,000}{2 \times 3,14 \times 50 \times 1,2} = 2,654 \Omega$$

$X_{c2}$  atau Kapasitor 2  $\mu\text{F}$  :

$$X_{c2} = \frac{1,000,000}{2 \times 3,14 \times 50 \times 2} = 1,592 \Omega$$

$X_{c3}$  atau Kapasitor 2,5  $\mu\text{F}$  :

$$X_{c3} = \frac{1,000,000}{2 \times 3,14 \times 50 \times 2,5} = 1,225 \Omega$$

$X_{c \text{ output}}$  atau kapasitor output 1  $\mu\text{F}$  :

$$X_{c \text{ output}} = \frac{1,000,000}{2 \times 3,14 \times 50 \times 1} = 3,185 \Omega$$

Dari hasil perhitungan untuk menentukan reaktansi kapasitif dari setiap kapasitor, terlihat bahwa pada kapasitor berukuran 1,2  $\mu\text{F}$ , terdapat hambatan sebesar 2,654  $\Omega$ . Sementara itu, kapasitor berukuran 2 $\mu\text{F}$  menunjukkan hasil hambatan sebesar 1,592  $\Omega$ , sedangkan kapasitor dengan kapasitansi 2,5 $\mu\text{F}$  menampilkan hambatan sebesar 1,225  $\Omega$ . Untuk kapasitor output berukuran 1 $\mu\text{F}$ , hasil perhitungan menunjukkan hambatan sebesar 3,185  $\Omega$ . Kemudian mencari hambatan dari kapasitor masing – masing *channel relay* yang diserikan dengan kapasitor output atau kapasitor 1  $\mu\text{F}$  :

Seri Kapasitor 1,2  $\mu\text{F}$  dan Kapasitor output 1  $\mu\text{F}$  :

$$X_{c \text{ seri } 1} = X_{c1} + X_{c \text{ out}} = 2,654 + 3,185 = 5,839 \Omega$$

Seri kapasitor 2  $\mu\text{F}$  dan kapasitor output 1  $\mu\text{F}$  :

$$X_{c \text{ seri } 2} = X_{c2} + X_{c \text{ output}} = 1,592 + 3,185 = 4,777 \Omega$$

Seri kapasitor 2,5  $\mu\text{F}$  dan kapasitor output 1  $\mu\text{F}$  :

$$Xc \text{ seri } 3 = Xc3 + Xc \text{ output} = 1,225 + 3,185 = 4,410 \Omega$$

Berdasarkan hasil perhitungan di atas, upaya dilakukan untuk menemukan nilai hambatan dari setiap kapasitor pada relay yang dihubungkan secara seri dengan kapasitor output atau kapasitor sebesar 1uF. Hasil analisis menunjukkan bahwa kapasitor seri pertama menampilkan hambatan sebesar 5,837  $\Omega$ , sedangkan kapasitor seri kedua memiliki hambatan sebesar 4,777  $\Omega$ , dan kapasitor seri ketiga menunjukkan hambatan sebesar 4,410  $\Omega$ . Perhitungan ini memberikan gambaran tentang karakteristik hambatan masing – masing kapasitor dalam rangkaian tersebut.

Setelah itu, menentukan tegangan pada ketiga kapasitor yang dihubungkan secara seri dengan kapasitor output atau kapasitor 1  $\mu$ F :

Mencari Tegangan Xc seri 1 :

$$Xc_{seri1} = \frac{220 \times Xc \text{ out}}{Xc \text{ seri } 1} = \frac{220 \times 3,185}{5,832} = 120 \text{ Volt}$$

$$Xc1 = \frac{220 \times Xc1}{Xc \text{ seri } 1} = \frac{220 \times 2,654}{5,832} = 100 \text{ Volt}$$

Mencari Tegangan Xc seri 2 :

$$Xc_{seri2} = \frac{220 \times Xc \text{ out}}{Xc \text{ seri } 2} = \frac{220 \times 3,185}{4,777} = 146 \text{ Volt}$$

$$Xc2 = \frac{220 \times Xc \text{ out}}{Xc \text{ seri } 2} = \frac{220 \times 1,592}{4,777} = 73 \text{ Volt}$$

Mencari Tegangan Xc seri 3 :

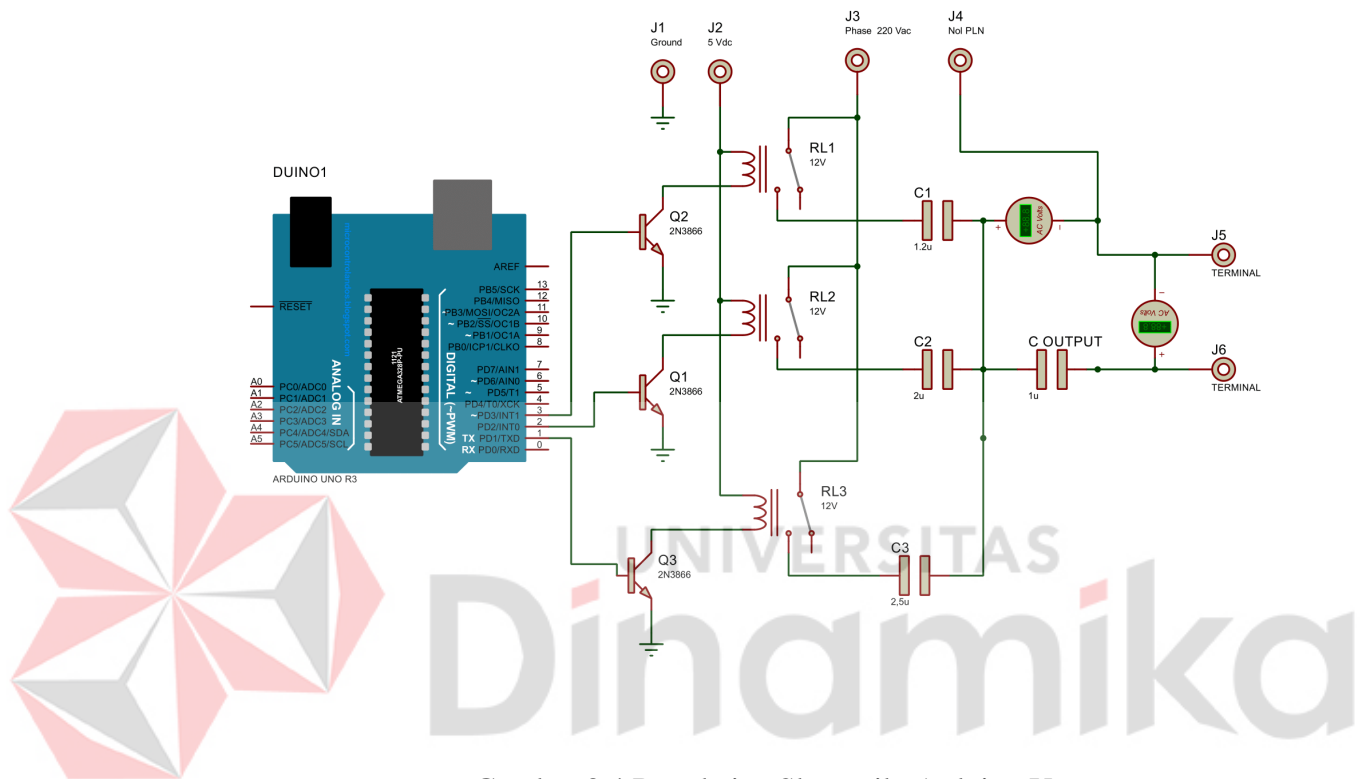
$$Xc_{seri3} = \frac{220 \times Xc \text{ out}}{Xc \text{ seri } 3} = \frac{220 \times 3,185}{4,410} = 159 \text{ Volt}$$

$$Xc3 = \frac{220 \times Xc \text{ out}}{Xc \text{ seri } 1} = \frac{220 \times 1,225}{4,410} = 61 \text{ Volt}$$

Melalui perhitungan tegangan pada kapasitor seri, ditemukan bahwa kapasitor seri 1 menghasilkan 120Volt, namun jika tidak diseri dengan kapasitor 1uF, tegangannya menjadi 100 Volt. Pada kapasitor seri 2, terlihat bahwa tegangan yang dihasilkan adalah 146 Volt, tetapi tanpa diseri dengan kapasitor 1uF, tegangannya turun menjadi 73Volt. Selanjutnya,



kapasitor seri 3 memberikan hasil tegangan sebesar 159 Volt, dan tanpa diseri dengan kapasitor 1 $\mu$ F, tegangannya menjadi 61 Volt. Fungsi utama dari penyusunan kapasitor secara seri ini adalah untuk membagi tegangan dari sumber listrik AC sebesar 220 Volt.



Gambar 3.4 Rangkaian Skematik Arduino Uno

Dari skema Gambar 3.4 terlihat bahwa terdapat suatu rangkaian yang terdiri dari beberapa *channel relay* yang terhubung ke *port* pada mikrokontroler Arduino Uno. Setiap *relay* pada *channel* tersebut kemudian dilengkapi dengan kapasitor berukuran berbeda untuk mengatur kecepatan kipas angin. Seperti pada *channel* satu terdapat kapasitor dengan kapasitansi 1.2  $\mu$ F, *channel* dua dilengkapi dengan kapasitor berukuran 2  $\mu$ F, dan *channel* ketiga memiliki kapasitor berukuran 2.5  $\mu$ F. Kapasitor-kapasitor ini nantinya akan dihubungkan secara seri dengan kapasitor output berukuran 1  $\mu$ F. Fungsi kapasitor output adalah sebagai pembagi tegangan yang kemudian dialirkan ke kipas angin. Voltmeter digunakan untuk memonitor tegangan pada kapasitor – kapasitor yang dihubungkan secara seri. Dengan konfigurasi ini, rangkaian diatur untuk mengontrol kecepatan kipas angin dengan memanfaatkan kapasitor-kapasitor berbeda pada setiap *channel* relay.

### 3.2 Instalasi *Environment*

Pada tahap instalasi *environment* terbagi 2 bagian terdapat beberapa *library – library* yang akan digunakan dalam proses klasifikasi suara seperti TensorFlow Hub, *soundfile*, *numpy*, *pandas* dan *library* lainnya yang dibutuhkan pada saat proses *training dataset*.

```
import random
import os
import glob
import time

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import librosa
from IPython import display
import soundfile as sf

import tensorflow as tf
import tensorflow_hub as hub
from tensorflow.keras import layers, Sequential
from tensorflow.keras.utils import plot_model

from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, precision_recall_fscore_support
from sklearn.metrics import accuracy_score, f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import plot_roc
```

TensorFlow Hub adalah salah satu *library* dari TensorFlow yang menyediakan berbagai model yang dapat diakses dan digunakan dengan mudah. Pada penelitian ini, *library* tersebut terdapat model YAMNet yang akan menjadi fokus utama. Untuk memanggil model tersebut, digunakan metode memuat link "<https://tfhub.dev/google/yamnet/1>". TensorFlow Hub mempermudah akses terhadap model *pre-trained* yang telah diunggah ke platform

tersebut, memungkinkan penelitian ini untuk memanfaatkan arsitektur dan pembelajaran yang telah dioptimalkan sebelumnya. Dengan memanfaatkan YAMNet melalui TensorFlow Hub, penelitian ini dapat dengan cepat dan efisien mengintegrasikan kemampuan klasifikasi audio yang canggih ke dalam sistem yang sedang dikembangkan.

```
Yamnet_url = 'https://tfhub.dev/google/yamnet/1'  
model_name = 'yamnet_1'  
set_trainable = False  
  
yamnet = get_tfhub_model(  
    yamnet_url, model_name,  
    model_trainable = set_trainable  
)
```

Kemudian pada tahap pengujian menggunakan fitur rekam suara *realtime* terdapat beberapa *library* yang sama seperti pada tahap *training* namun lebih sedikit dan terdapat *library* baru seperti *Serial*. *Serial* merupakan *library* yang digunakan untuk menghubungkan ke mikrokontroler Arduino Uno dari Python. Pemasangan *library* - *library* tersebut dilakukan dengan cara menginstal python sesuai versi yang dapat terintegrasi *library* yang akan digunakan dan plugin yang dibutuhkan pada terminal pip atau anaconda ataupun bisa juga menggunakan Visual Studio Code. Arduino Uno yang didefinisikan terdapat String terminal *Read*, dipakai untuk pembacaan data dari *Serial* terminal Arduino yang memakai tipe data String.

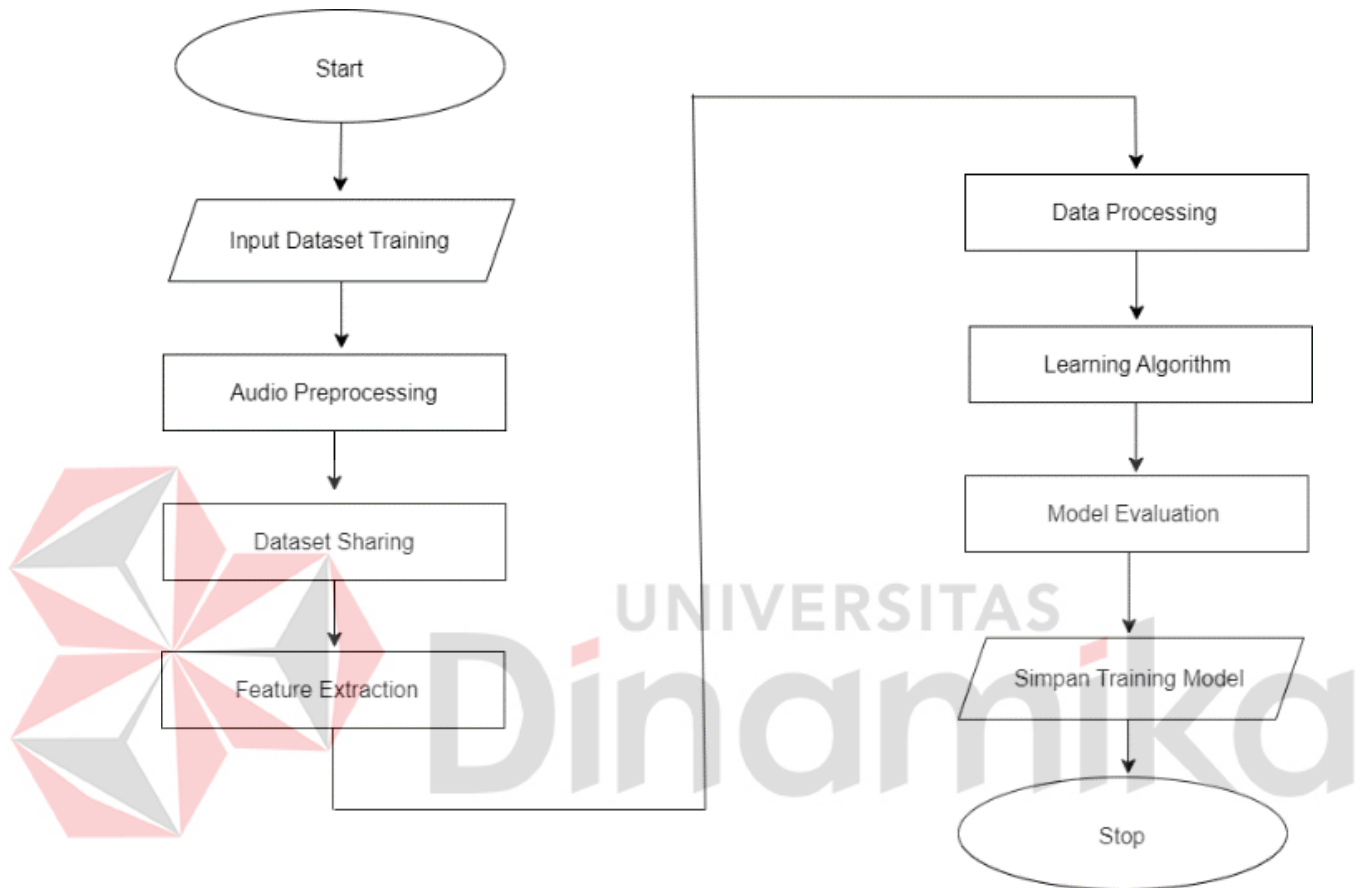
```
import sounddevice as sd  
import numpy as np  
from scipy.io.wavfile import write, read  
from tensorflow.keras.models import load_model  
import serial  
import time  
import platform
```

### 3.3 Data Pre – paration

#### 3.3.1 Flowchart Dataset

Pada penelitian Tugas Akhir ini penulis menggunakan audio *dataset* yang diciptakan oleh peneliti berdasarkan perintah – perintah untuk mengatur kipas angin menggunakan tiga

bahasa seperti Bahasa Indonesia, Bahasa Inggris, dan Bahasa Jawa. Dalam pembuatan audio *dataset* terdapat beberapa langkah pembuatannya.



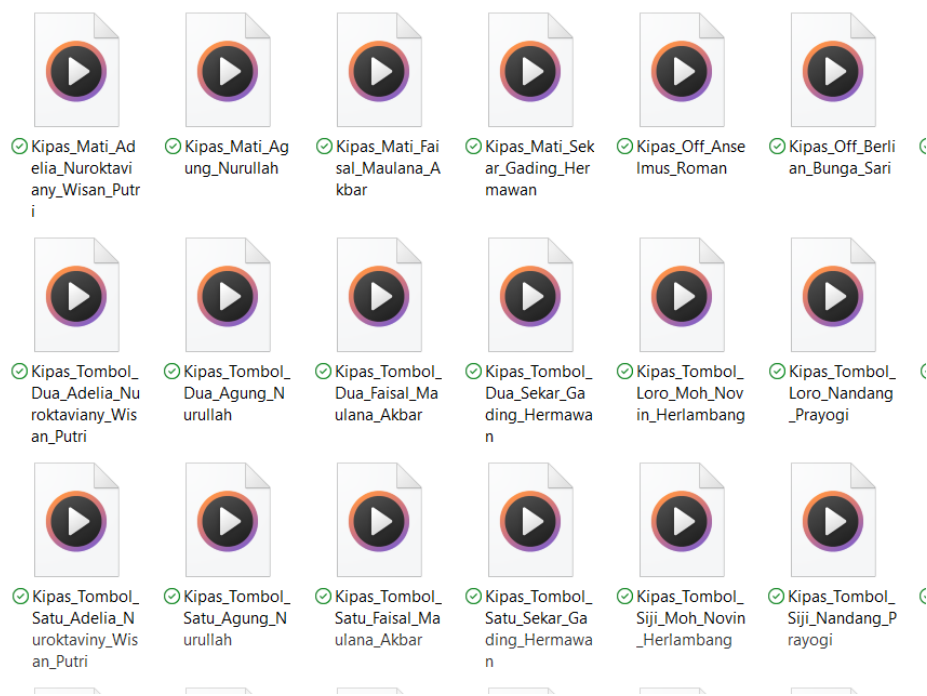
Gambar 3.6 *Flowchart* audio dataset

Pertama - tama pengumpulan data audio yang sesuai tujuan yang diperintahkan pada penelitian ini. Data ini berupa suara manusia ataupun suara yang mewakili perintah yang ingin dikenali. Kemudian memberikan *label* disetiap data audio supaya dapat mewakili setiap rekaman. Selanjutnya pemrosesan data perlu dilakukan data audio sebelum digunakan, hal ini bisa mengubah format audio seperti WAV atau MP3, normalisasi *volume* atau menghilangkan kebisingan latar belakang. Dalam pembagian *dataset* dapat menjadi set pelatihan, set validasi, dan set pengujian. Set pelatihan dipakai untuk melatih model, set validasi dipakai untuk mengoptimasi parameter model dan set pengujian digunakan untuk

mengukur kinerja model. Lalu data audio perlu mengubah menjadi fitur numerik yang bisa dimasukkan ke dalam model. Model tersebut bisa berupa representasi *spectral* seperti *Mel – Frequency Cepstral Coefficients* (MFCC). Selanjutnya latih model tersebut menggunakan data pelatihan yang sudah diproses. Selama proses latihan, model akan belajar mengenali pola – pola dalam data suara. Kemudian tahap validasi untuk menyetel parameter model data, seperti jumlah lapisan pada jaringan saraf. Setelah itu gunakan set pengujian untuk mengevaluasi kemampuan model tersebut. Hitung *matrix* seperti akurasi, presisi, *recall* dan *f1-score* berfungsi sebagai mengukur seberapa baik model yang telah dibuat melakukan pengenalan suara. Setelah dilakukan pemodelan jika model data tersebut memiliki kinerja yang baik, maka dapat digunakan untuk melakukan pengenalan suara pada data baru.

### 3.3.2 Pengumpulan Dataset

Audio dataset yang digunakan pada penelitian ini merupakan suara manusia yang mengucapkan perintah menggunakan tiga bahasa. Dalam setiap perintah menggunakan kata kunci “kipas” untuk perintah kipas mati, kipas *off*, kipas pejah dan “kipas tombol” untuk perintah dari setiap masing- masing *level* kipas angin seperti kipas tombol satu, kipas tombol siji, kipas tombol *one*, kipas tombol dua, kipas tombol *two*, kipas tombol loro, kipas tombol tiga, kipas tombol *three*, kipas tombol telu. Pada saat penulis mengumpulkan dataset terhitung total 270 *file* audio dari 18 dengan 4 perintah dari setiap masing – masing bahasa dengan 5 kali pengucapan dari setiap masing – masing perintah ke dalam bentuk *file* audio baru.



Gambar 3.7 Pengumpulan Dataset

### 3.3.3 Augmentasi Data

Dalam tahap augmentasi data pada penelitian ini, peneliti mengembangkan strategi peningkatan data untuk menyempurnakan kumpulan data audio yang sudah digunakan. Teknik yang dikenal sebagai “augmentasi data” digunakan untuk memperbesar ukuran kumpulan data. Akibatnya model yang digunakan untuk menghitung keluaran model akan memiliki varian yang lebih tinggi dibandingkan masukannya. Penulis melakukan augmentasi audio dengan menerapkan saluran *high pass*, *medium pass* dan *low pass* pada sampel suara yang terkumpul.

Augmentasi dengan *channel high pass*, *medium pass*, dan *low pass* memberikan variasi frekuensi pada dataset suara berbeda dengan filter *low pass* yang menjamin frekuensi konstan, *high pass* digunakan untuk memastikan frekuensi yang konsisten di udara. Saluran *medium pass* menciptakan kesan tengah antara keduanya. Hal tersebut bertujuan untuk membuat model lebih adaptif terhadap variasi lingkungan akustik yang mungkin terjadi dalam kehidupan sehari – hari. Untuk melakukan generalisasi kinerja model yang digunakan dalam penelitian ini, perlu menggunakan data tambahan.

Dengan menerapkan augmentasi informasi suara melalui saluran *high pass*, *medium pass*, dan *low pass*, penulis berupaya meningkatkan ketangguhan dan generalisasi model terhadap variasi yang mungkin terjadi dalam keadaan lingkungan yang berbeda. Komposisi ini diharapkan dapat membantu model untuk lebih baik dalam mengenali dan memahami berbagai varian suara yang mungkin dipahami dalam implementasi praktis sistem klasifikasi suara yang sedang penulis kembangkan. Sehingga total akhir keseluruhan pengumpulan data yang telah dilakukan tahap augmentasi sebanyak 720 file yang terdiri dari 4 perintah dari 3 bahasa.

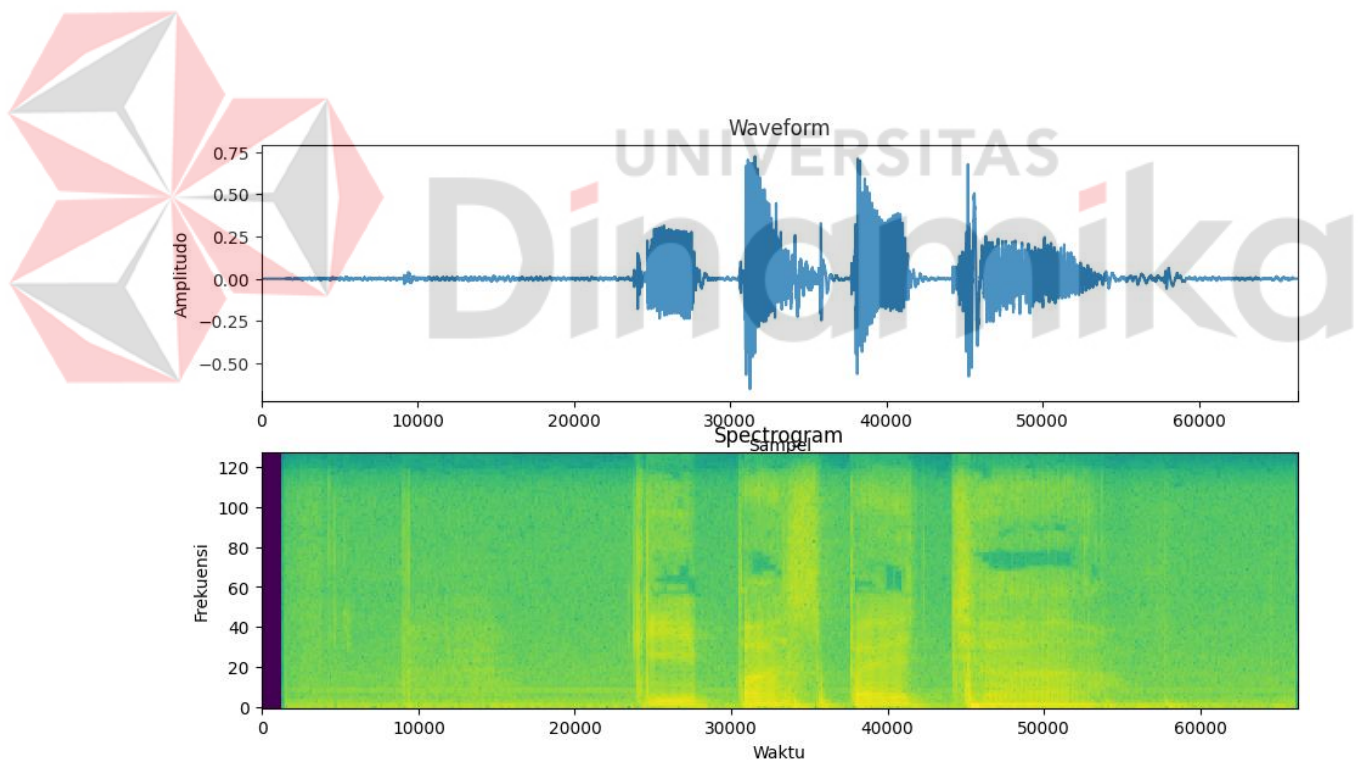


```
File Kipas_Mati (10).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (10)_1.wav
File Kipas_Mati (10).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (10)_2.wav
File Kipas_Mati (10).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (10)_3.wav
File Kipas_Mati (10).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (10)_4.wav
File Kipas_Mati (10).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (10)_5.wav
File Kipas_Mati (11).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (11)_1.wav
File Kipas_Mati (11).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (11)_2.wav
File Kipas_Mati (11).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (11)_3.wav
File Kipas_Mati (11).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (11)_4.wav
File Kipas_Mati (11).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (11)_5.wav
File Kipas_Mati (12).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (12)_1.wav
File Kipas_Mati (12).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (12)_2.wav
File Kipas_Mati (12).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (12)_3.wav
File Kipas_Mati (12).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (12)_4.wav
File Kipas_Mati (12).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (12)_5.wav
File Kipas_Mati (13).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (13)_1.wav
File Kipas_Mati (13).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (13)_2.wav
File Kipas_Mati (13).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (13)_3.wav
File Kipas_Mati (13).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (13)_4.wav
File Kipas_Mati (13).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (13)_5.wav
File Kipas_Mati (14).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (14)_1.wav
File Kipas_Mati (14).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (14)_2.wav
File Kipas_Mati (14).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (14)_3.wav
File Kipas_Mati (14).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (14)_4.wav
File Kipas_Mati (14).wav has been augmented and saved to ./dataset/augmentation/Kipas_Mati (14)_5.wav
...
File Kipas_Tombol_Two.wav has been augmented and saved to ./dataset/augmentation/Kipas_Tombol_Two_2.wav
File Kipas_Tombol_Two.wav has been augmented and saved to ./dataset/augmentation/Kipas_Tombol_Two_3.wav
File Kipas_Tombol_Two.wav has been augmented and saved to ./dataset/augmentation/Kipas_Tombol_Two_4.wav
File Kipas_Tombol_Two.wav has been augmented and saved to ./dataset/augmentation/Kipas_Tombol_Two_5.wav
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Gambar 3.8 Augmentasi Audio Dataset

### 3.3.4 Karakteristik Audio

Pada langkah ini sangat penting terhadap karakteristik audio yang digunakan. Dengan proses pemantauan secara visual menggunakan *waveform* dan *spectrogram* melibarkan representasi visual dari sinyal audio, yang esensial untuk pemahaman mendalam terhadap ciri – ciri unik yang terkandung dalam data audio. Melalui visual *waveform*, pemantauan terhadap fluktuasi *amplitude* terhadap waktu dapat dilakukan dengan jelas. Ini memberikan gambaran tentang dinamika dasar, termasuk durasi, tempo, dan varisasi *amplitude* yang mungkin memengaruhi karakteristik audio secara keseluruhan.



Gambar 3.9 *Waveform* dan *Spectrogram*

Sementara itu, *spectrogram* memberikan informasi lebih lanjut dengan mempresentasikan distribusi energi frekuensi dalam sinyal audio terhadap waktu. Dengan menggunakan plot spektrum frekuensi ini, pemantauan terhadap perubahan frekuensi, *pitch*,



dan elemen – elemen harmonis dapat dicapai. Kombinasi visual *waveform* dan *spectrogram* menciptakan pendekatan yang komprehensif untuk menganalisis dan memahami karakteristik audio. Tahapan ini memainkan peran kunci dalam persiapan data, memungkinkan penelitian ini untuk mengenali pola – pola yang mendasari data audio sebelum memasuki tahap *encoding* dan pengembangan model. Dengan demikian, pemantauan visual ini pada *waveform* dan *spectrogram* menjadi langkah awal yang krusial seperti pentingnya pemantauan visual tersebut sebagai fondasi atau dasar yang sangat penting untuk langkah – langkah dalam perjalanan analisis audio yang lebih mendalam dalam kerangka penelitian Tugas Akhir ini.

### 3.3.5 *Rename File Dataset*

Setelah augmentasi data, langkah berikutnya adalah melakukan *rename* pada setiap *file* audio dataset. Penulis menerapkan strategis *rename file* dengan menambahkan kata kunci perintah dan nomor urut *file* pada setiap sampel suara. Sebagai contoh, file suara yang mempresentasikan kategori “kipas mati” akan diberi nama “kipas\_mati (1)” dan demikiran seterusnya. Begitu juga dengan perintah lainnya, seperti “kipas\_tombol\_satu (1)”, “kipas\_tombol\_dua (1)” dan “kipas\_tombol\_tiga (1)”. Proses *rename* ini dilakukan memudahkan pengelolaan dan pengenalan klasifikasi dari setiap sampel suara selama proses pelatihan model.

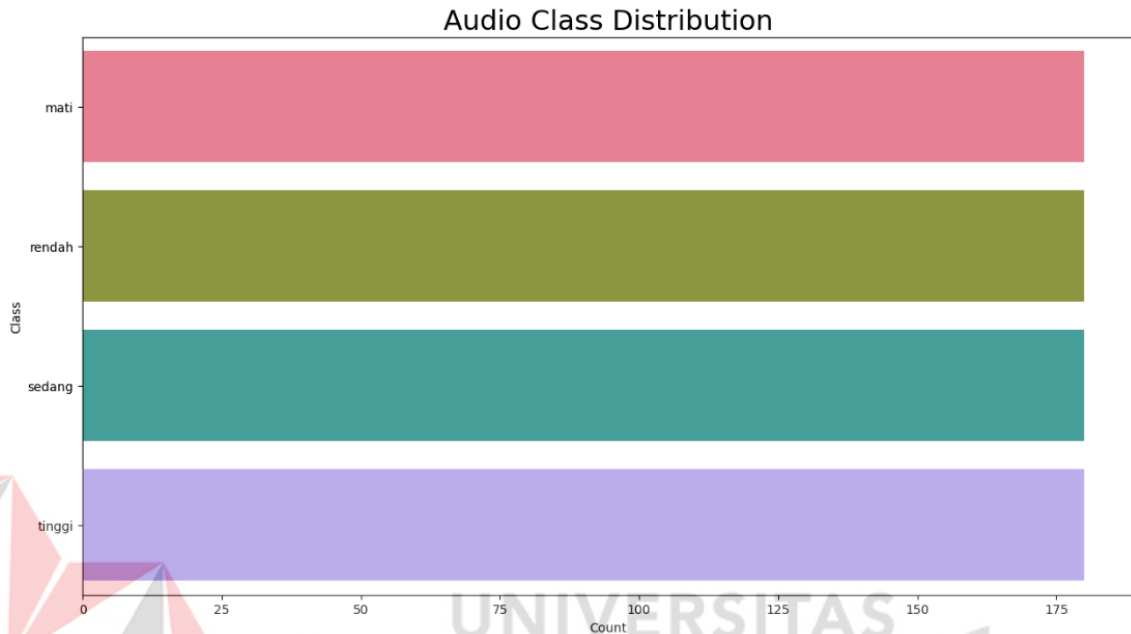
Dengan menggunakan metode *rename* ini, peneliti memastikan bahwa setiap file dalam dataset memiliki label yang jelas dan dapat diidentifikasi. Hal ini akan membantu efisiensi dan keakuratan dalam membangun model pengenalan suara, karena setiap sampel suara akan terkait dengan perintah dan nomor urut yang konsisten. Proses *rename file* dataset ini merupakan langkah kritis dalam persiapan dataset sebelum dilakukan tahap pelatihan dan pengujian model.



Gambar 3.10 **Rename** File Audio Dataset

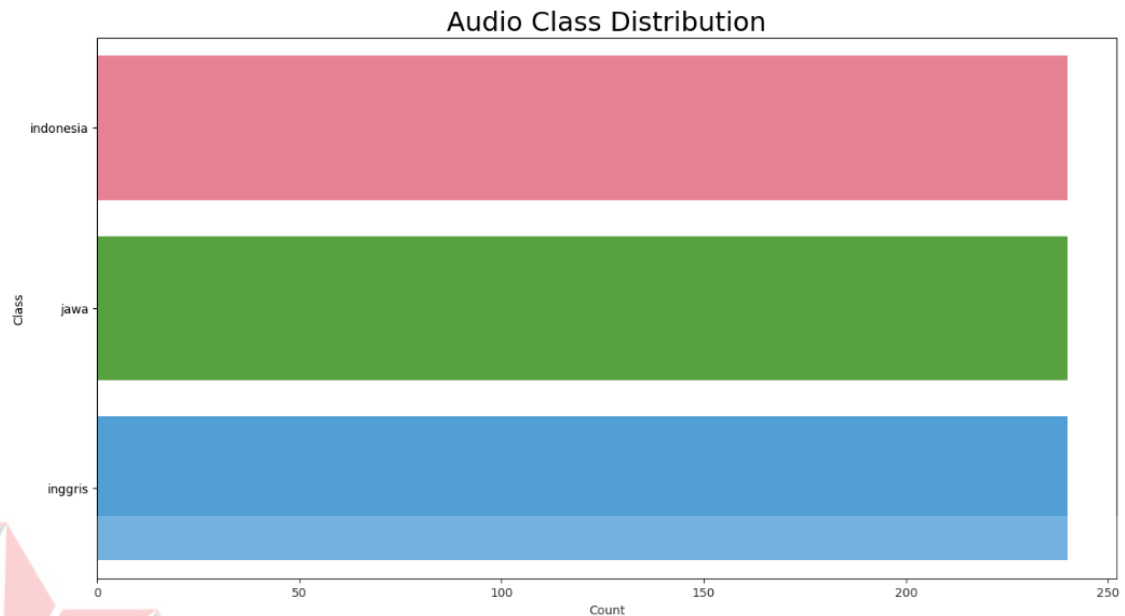
### 3.3.6 Pengelompokan Data

Setelah selesai tahap *rename* semua file dataset, langkah selanjutnya adalah melakukan pengelompokan data ke dalam folder 4 kategori kecepatan kipas angin seperti mati, rendah, sedang dan tinggi. Dengan pengelompokan ini, peneliti dapat memfasilitasi proses pelatihan dan evaluasi model untuk setiap kategori suara dengan lebih terstruktur.



Gambar 3.11 Pengelompokan Data Berdasarkan Kategori

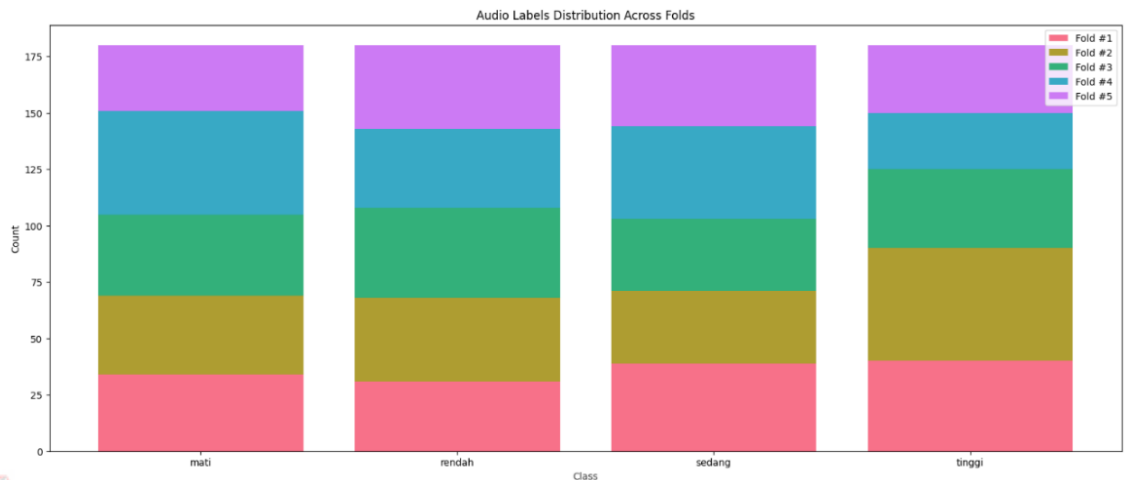
Kemudian dalam masing-masing *folder* kategori, data audio kemudian dikelompokkan berdasarkan bahasa. Peneliti membuat *subfolder* untuk setiap bahasa yang relevan, yaitu Bahasa Indonesia, Bahasa Jawa dan Bahasa Inggris. Hal tersebut bertujuan untuk menciptakan struktur yang lebih rinci dan terorganisir, memudahkan pengelolaan dataset untuk setiap *level* kecepatan kipas angin menggunakan bahasa yang digunakan dalam pengucapan perintah. Dengan pendekatan pengelompokan ini, penulis memastikan bahwa model klasifikasi suara dapat mengakomodasi berbagai bahasa yang mungkin digunakan dalam penggunaan praktis, sehingga meningkatkan fleksibilitas dan generalisasi model tersebut.



Gambar 3.12 Pengelompokan Data Berdasarkan Bahasa

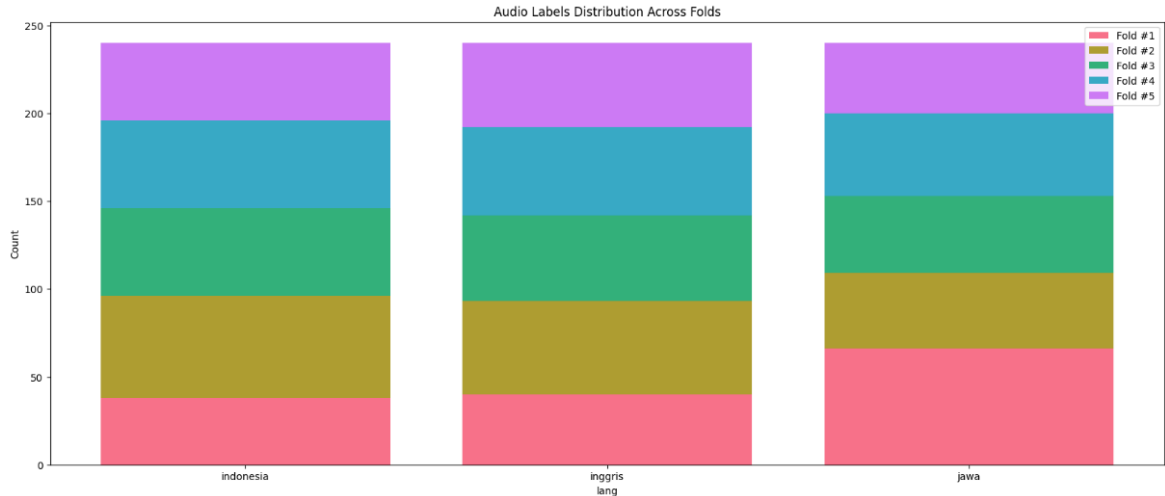
### 3.3.7 Distribusi Data Kategori

Setelah proses *rename* dataset selesai, langkah berikutnya adalah mendistribusikan dataset ke dalam lima folder yang dinamai sebagai *folds*, yaitu *folds 1*, *folds 2*, *folds 3*, *folds 4*, dan *folds 5*. Distribusi ini dilakukan sesuai dengan datasheet yang telah dibuat secara random oleh program. Tujuan utama dari distribusi ke dalam *folds* ini adalah untuk memastikan bahwa setiap *folds* berisi representasi yang seimbang dari berbagai bahasa dan kategori perintah level kecepatan kipas angin. Hal ini penting untuk mencapai hasil evaluasi dan validasi model yang lebih objektif.



Gambar 3.13 Distribusi Folds Berdasarkan *Label*

Dalam setiap *folds*, dataset akan terdiri dari kategori *level* “mati”, “rendah”, “sedang” dan “tinggi”. Setiap kategori ini mempresentasikan tingkat intensitas suara yang berbeda. pengelompokan ini dijalankan untuk memastikan bahwa setiap *folds* mencakup keragaman level kecepatan kipas angin yang mungkin ditemui dalam penggunaan praktis. Dengan mendistribusikan dataset ke dalam *folds* secara proporsional, peneliti berharap dapat mencapai generalisasi model yang lebih baik dan mampu mengenali variasi suara yang lebih luas. Proses ini adalah bagian sangat penting dari persiapan dataset sebelum masuk ke tahap pelatihan dan evaluasi model.



Gambar 3.14 Distribusi *Folds* berdasarkan Bahasa

### 3.3.8 Datasheet

Dalam tahap pembuatan datasheet, peneliti menyusun tabel *datasheet* yang mencakup beberapa atribut penting untuk setiap sampel suara dalam dataset. Setiap baris tabel ini mencantumkan informasi yang mencakup “*slice file name*” yaitu nama *file* yang dihasilkan setelah proses augmentasi dan rename, “*fsID*” yang merupakan identifikasi unit untuk setiap *file* audio, “*start*” dan “*end*” yang menunjukkan batas awal dan akhir dari masing – masing *slice audio*, serta “*salience*” yang mencerminkan tingkat kepentingan dari setiap *slice*. Selain itu, atribut “*fold*” digunakan untuk menunjukkan ke dalam fold mana suara tersebut didistribusikan selama proses validasi atau evaluasi model. “*classID*” dan “*class*” mencakup informasi tentang kategori *level* kecepatan kipas angin, seperti “*mati*”, “*rendah*”, “*sedang*”, dan “*tinggi*”. Terakhir, atribut “*language*” mencatat bahasa yang digunakan dalam perintah yang direkam.

Setelah menyusun tabel dengan atribut tersebut, datasheet ini kemudian disimpan dalam format *Comma-Separated Values* (CSV). Dengan menyimpan datasheet dalam format ini, memudahkan akses dan pengolahan data menggunakan berbagai alat analisis data, sehingga mempercepat proses evaluasi dan pelatihan model dalam penelitian ini. Proses pembuatan datasheet dan penyimpanannya dalam format CSV menjadi langkah penting dalam pengelolaan dataset yang akan digunakan dalam tahap selanjutnya dari penelitian ini.

	slice_file_name	fsID	start	end	saliency	fold	classID	class	lang
0	kipas_mati_0001.wav	194197	43	53	1	4	0	mati	indonesia
1	kipas_mati_0002.wav	942852	26	32	1	4	0	mati	indonesia
2	kipas_mati_0003.wav	973538	21	25	1	5	0	mati	indonesia
3	kipas_mati_0004.wav	286614	55	58	1	5	0	mati	indonesia
4	kipas_mati_0005.wav	398213	69	78	1	2	0	mati	indonesia
5	kipas_mati_0006.wav	415443	94	96	1	5	0	mati	indonesia
6	kipas_mati_0007.wav	514843	1	5	1	1	0	mati	indonesia
7	kipas_mati_0008.wav	293990	36	42	1	2	0	mati	indonesia
8	kipas_mati_0009.wav	232791	55	63	1	3	0	mati	indonesia
9	kipas_mati_0010.wav	649273	78	87	1	4	0	mati	indonesia
10	kipas_mati_0011.wav	839319	46	48	1	4	0	mati	indonesia
11	kipas_mati_0012.wav	243489	83	85	1	5	0	mati	indonesia
12	kipas_mati_0013.wav	359819	63	68	1	5	0	mati	indonesia
13	kipas_mati_0014.wav	975481	19	21	1	4	0	mati	indonesia
14	kipas_mati_0015.wav	314052	64	65	1	1	0	mati	indonesia
15	kipas_mati_0016.wav	516767	16	23	1	5	0	mati	indonesia
16	kipas_mati_0017.wav	837826	52	62	1	3	0	mati	indonesia
17	kipas_mati_0018.wav	277062	30	38	1	4	0	mati	indonesia
18	kipas_mati_0019.wav	863752	65	70	1	5	0	mati	indonesia
19	kipas_mati_0020.wav	997128	88	97	1	1	0	mati	indonesia

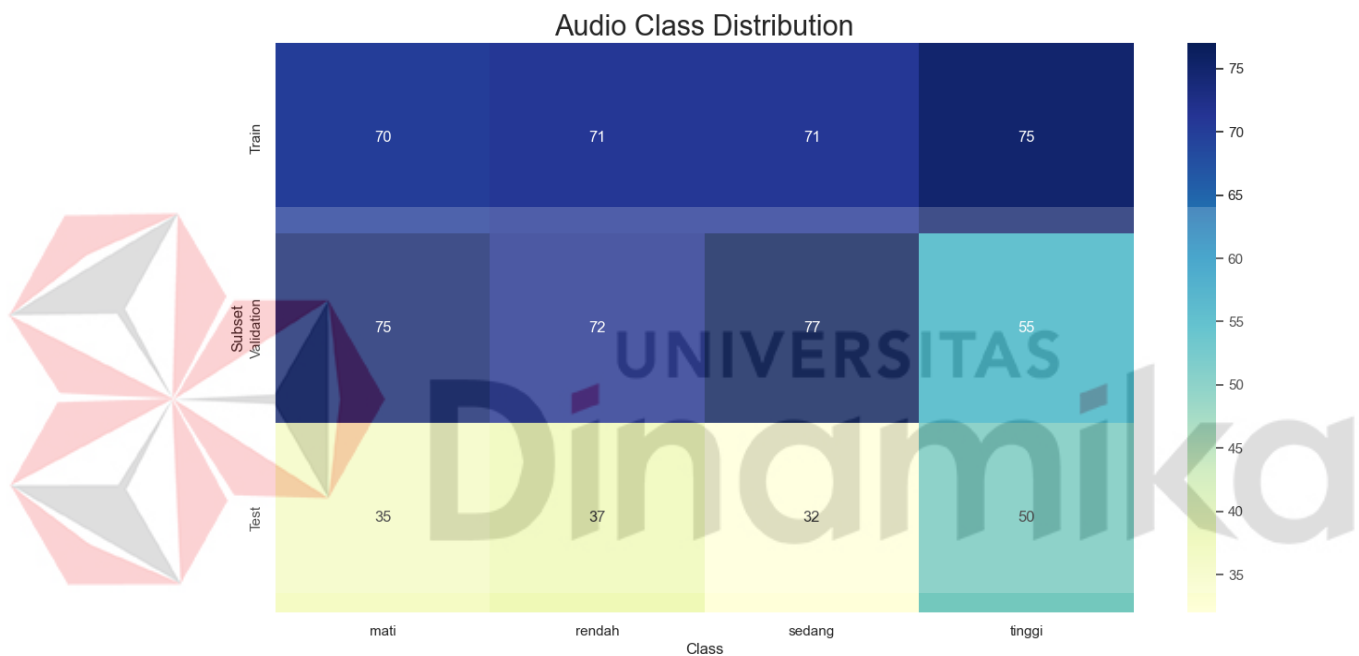
Gambar 3.15 *Datasheet*

### 3.3.9 Distribusi *Folds Subset*

Pada tahap distribusi *folds* melakukan pembagian dataset ke dalam tiga subset utama: *train*, *test* dan *validation*. *Fold 1* dan *3* dipilih untuk menjadi bagian dari *train data*, sedangkan *4* dan *5* dialokasikan sebagai *validation data*. Terakhir, *folds 2* ditetapkan sebagai *test data*. Pembagian ini bertujuan untuk mengoptimalkan pelatihan model dengan menggunakan *train data*, menguji kinerja model secara objektif menggunakan *test data* yang belum pernah digunakan sebelumnya, dan memvalidasi hasil model pada *validation data* untuk mencegah *overfitting*.

Pembagian ini dilakukan dengan cermat untuk memastikan bahwa setiap *subset* mencakup representasi yang seimbang dari seluruh dataset, termasuk berbagai bahasa dan kategori *level* kecepatan kipas angin. Dengan demikian, model yang dihasilkan dapat memahami dan mengenali suara secara efektif dalam berbagai kondisi. Proses pembagian

ini dilakukan dengan memperhatikan proporsi dataset yang memadai untuk memastikan bahwa model diberikan variasi yang memadai untuk melatih dan menguji kemampuannya. Langkah distribusi folds ini memainkan peran penting dalam persiapan dataset sebelum melanjutkan ke tahap pelatihan, evaluasi dan validasi model klasifikasi suara.



Gambar 3.16 Distribusi *Folds* Subset

Dari masing – masing *subset* memiliki jumlah dataset yang berbeda pada *subset train* memiliki 287 audio dataset, kemudian pada *validation* terdapat 279 audio dataset dan *test* terdapat 154 audio dataset sehingga dari total keseluruhan dari 3 subset terdapat 720 audio datasets.



Dataset Split Sizes

	Subset	Number of Samples
0	Train	287
1	Validation	279
2	Test	154
3	Total	720

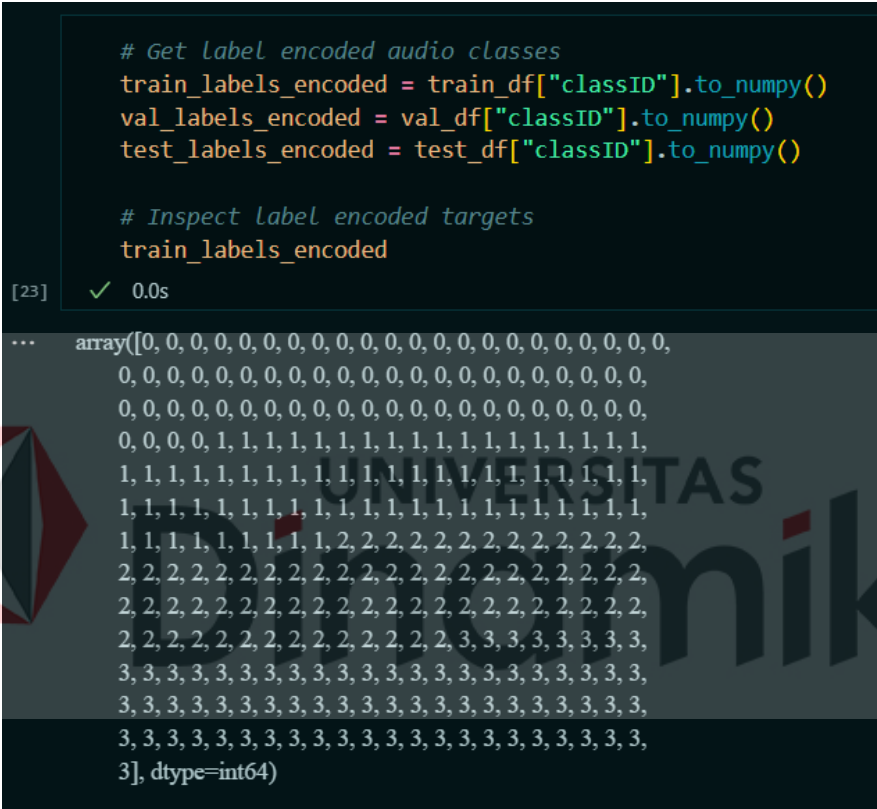
Gambar 3.17 Tabel *Subset* Dataset

### 3.3.10 Label Encoding

Lalu langkah berikutnya adalah melakukan *encoding* pada *label* audio yang bertujuan mengubah *class label* audio menjadi representasi numerik pada tiga subset yang terdiri dari *train*, *test*, dan *validation*. *Encoding* ini merupakan tahap krusial dalam pengolahan data audio yang memiliki tujuan utamanya adalah mengonversi informasi *class* audio menjadi numerik yang bisa dipahami oleh model pembelajaran mesin. Proses ini dilakukan pada setiap *subset* data, termasuk *train* untuk melatih model, *test* untuk menguji performa model dan *validation* untuk memvalidasi hasil pembelajaran.

Dalam konteks tersebut, *encoding label* audio menjadi representasi numerik memungkinkan model dapat memahami dan memproses informasi *class* dengan lebih efektif. Setiap *label* audio diberikan representasi numerik uniknya, menciptakan keseragaman dan keterbacaan data bagi model pembelajaran mesin. Proses ini merupakan bagian integral dari persiapan data sebelum dilibatkannya model dalam langkah pelatihan, pengujian dan validasi. dengan melakukan *encoding* ini, penelitian Tugas Akhir ini memastikan bahwa informasi kelas pada data audio dapat diintegrasikan secara efisien ke

dalam model pembelajaran mesin, memungkinkan model untuk mengenali dan memahami pola yang menjadi dasar dari ciri – ciri khas audio. Tahap – tahap ini menjadi dasar penting dalam upaya pengembangan model yang dapat memberikan hasil yang akurat dan handal dalam pengelompokan audio pada fase selanjutnya.



```

# Get Label encoded audio classes
train_labels_encoded = train_df["classID"].to_numpy()
val_labels_encoded = val_df["classID"].to_numpy()
test_labels_encoded = test_df["classID"].to_numpy()

# Inspect Label encoded targets
train_labels_encoded

```

[23] ✓ 0.0s

```

... array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3], dtype=int64)

```

Gambar 3.18 *Label Encoding*

### 3.3.11 *One – Hot Encoding*

Masih dalam tahap *encoding class* namun pada tahap ini dilakukan menggunakan metode *one – hot encoding* bertujuan untuk menerjemahkan representasi *label* data ke dalam vektor biner. Metode tersebut mempunyai fokus dalam penyederhanaan representasi kelas dalam konteks tugas *classification*. Saat melibatkan data audio pada *subset train, test, dan validation*, setiap *class* direpresntasikan oleh vektor biner dengan panjang sesuai dengan

jumlah *class* yang ada. Dalam konteks ini, elemen vektor memiliki nilai satu indikasi *class* yang diwakilkan, sementara elemen – elemen lainnya memiliki nilai nol.

Proses *one – hot encoding* ini memiliki tujuan fundamental dalam model pembelajaran mesin untuk memahami dan memproses informasi class dengan lebih efisien. Menggunakan representasi numerik ini, dapat mengenali hubungan antar *class*, memfasilitasi identifikasi pola yang mendasari karakteristik audio. Selain itu, metode ini membantu mencegah terjadinya kecondongan yang mungkin muncul dalam interpretasi model terhadap class tertentu, menciptakan interpretasi yang objektif dan konsisten. Dengan menerapkannya metode *one – hot encoding* pada langkah ini, penelitian tugas akhir ini memastikan bahwa data yang digunakan untuk pelatihan, pengujian, dan validasi model klasifikasi audio siap untuk diolah secara efektif, membawa pada hasil analisis yang handal pada tahap penelitian berikutnya.



```
# One-hot encode train labels
train_one_hot = one_hot_encode(train_labels_encoded, num_classes)

# One-hot encode validation labels
val_one_hot = one_hot_encode(val_labels_encoded, num_classes)

# One-hot encode test labels
test_one_hot = one_hot_encode(test_labels_encoded, num_classes)

# View one-hot encodings of train labels
train_one_hot

[26] ✓ 0.0s

... <tf.Tensor: shape=(287, 4), dtype=float32, numpy=
array([[1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       ...,
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.]], dtype=float32)>
```

Gambar 3.19 *One – Hot Encoding Label*

### 3.4 Data Training

Pada tahap *training* merupakan melatih model dengan menggunakan 2 skenario yang berbeda, satu dengan 12 *epoch* dan 15 *epoch*. Proses ini memiliki peran krusial dalam pengembangan model karena memungkinkan model untuk meningkatkan pemahaman terhadap data yang telah dipersiapkan dengan lebih baik. Dengan menerapkan proses pelatihan pada model dengan jumlah *epoch* yang bervariasi, dapat terevaluasi kemampuan model dalam menangkap pola – pola yang kompleks dari dataset pelatihan yang ada.

### 3.5 Model YAMNet

Setelah melakukan tahap persiapan dataset untuk melatih model, perhatian utama diberikan pada struktur *layer* yang akan digunakan. Proses ini melibatkan pemakaian Audio *Generator*, suatu alat yang digunakan untuk menghasilkan pasangan *embedding* dan *label* dari berkas audio dengan memberikan jalur dan label yang telah di encode menggunakan metode *one – hot encoding*. Selanjutnya, dibangun suatu alur masukan menggunakan API *tf.data* dengan cara memanfaatkan jalur dan *label* untuk data audio. Alur ini mencakup langkah – langkah seperti mendefinisikan tanda keluar dataset, menciptakan dataset *frame* dengan *generator*, menerapkan pengacakan berdasarkan kondisi tertentu, memberlakukan *batch – an*. Menerapkan *caching* dan *prefetching* berdasarkan kondisi.

Langkah selanjutnya adalah membaginya menjadi subset yang relevan untuk proses pelatihan, pengujian dan validasi. pembagian dataset ini krusial untuk menguji kinerja model dengan tepat. Subset pelatihan digunakan untuk melatih model, subset pengujian digunakan untuk mengukur kinerja model secara objektif, sementara subset validasi membantu menilai apakah model dapat umum digunakan. Dengan pembagian dataset tersebut, dapat dijamin bahwa model dapat menangkap pola dengan baik tanpa *overfitting*.

Setelah dataset terbagi, tahap berikutnya melibatkan penganalisisan bentuk sampel dan label untuk memastikan kesesuaian dengan model klasifikasi audio YAMNet. Inisialisasi model dimulai dengan mendefinisikan *kernel initializer* yang akan digunakan. Kemudian, pada *layer* input dibentuk dengan *shape* 4, 1024 yang merupakan *input* data empat dimensi, setiap dimensi memiliki Panjang 1024, dan tipe data yang digunakan adalah *tf.float32*. Pada *layer* LSTM, pendekatan *bidirectional* dengan 128 *initializer* digunakan untuk meningkatkan kemampuan model dalam menangkap konteks temporal dari data audio.

Di tahap ini, diterapkan dropout sebesar 0,25 pada *layer* audio LSTM. *Dropout* digunakan untuk mengurangi *overfitting* dengan secara acak mengabaikan sebagian unit selama pelatihan. Hal ini membantu meningkatkan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Penggunaan *dropout* pada *layer* audio LSTM bertujuan untuk mengoptimalkan kinerja model secara keseluruhan.

audio_input	input:	[(None, 4, 1024)]
InputLayer	output:	[(None, 4, 1024)]



bidirectional(lstm)	input:	(None, 4, 1024)
Bidirectional(LSTM)	output:	(None, 256)



drop out	input:	(None, 256)
Drop out	output:	(None, 256)



output_layer	input:	(None, 256)
Dense	output:	(None, 4)

Gambar 3.20 **Struktur** Model

Langkah terakhir adalah menghitung probabilitas klasifikasi pada model. proses ini memungkinkan model memberikan prediksi terhadap kategori – kategori yang telah diidentifikasi selama proses pelatihan. Dengan cara ini dapat mengevaluasi sejauh mana model mampu memahami dan mengenali pola audio yang bersifat unik. Dari keseluruhan

pada penelitian tugas akhir ini dirancang dengan hati – hati untuk memastikan bahwa model audio classification yang dibangun mampu memberikan hasil yang akurat dan dapat diandalkan dalam mengklasifikasikan data audio.

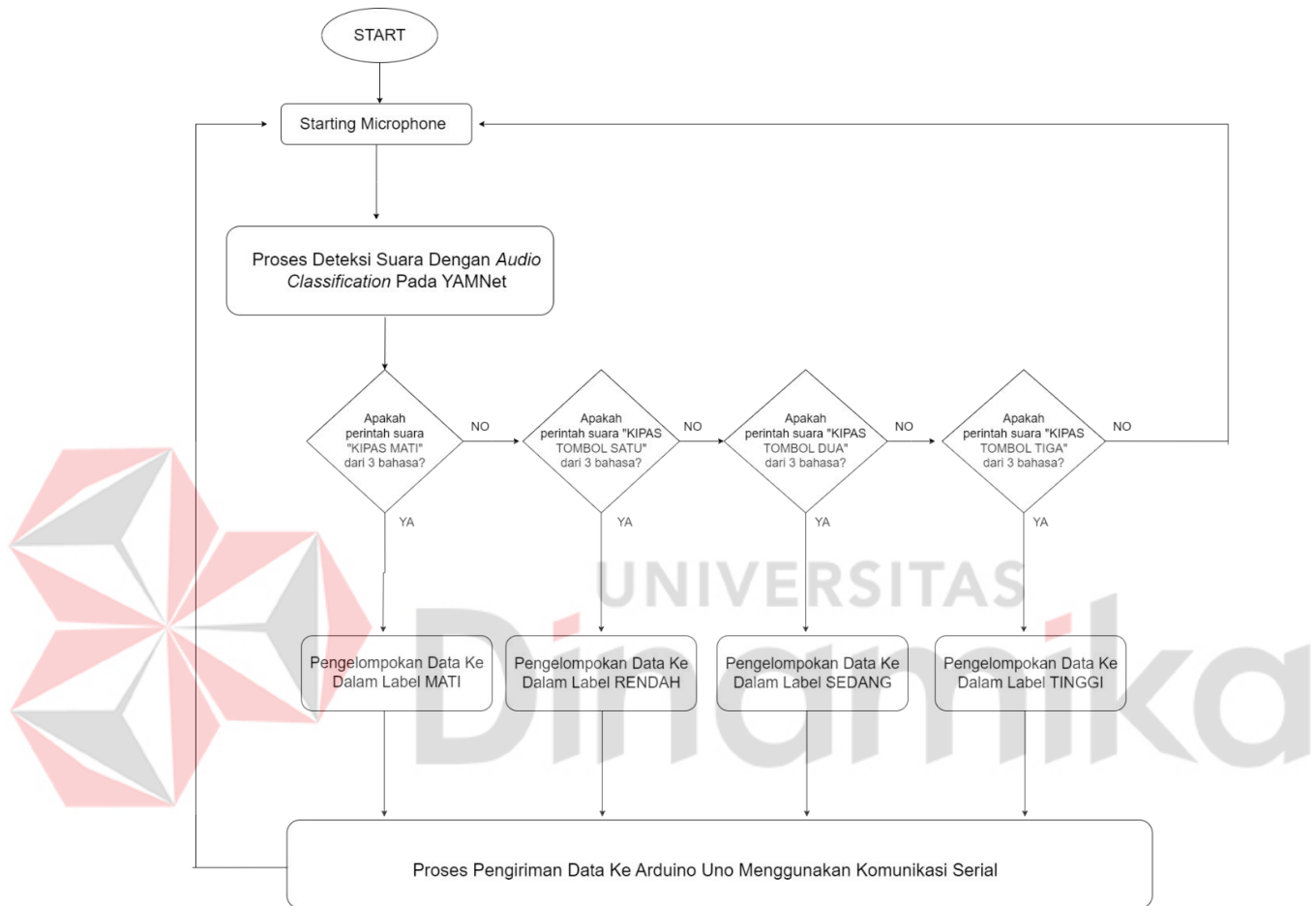
Model YAMNet dengan program yang telah dirancang melibatkan penggunaan datasheet dan model classifier berformat H5 yang merupakan hasil pelatihan dataset. Kemudian pengguna melakukan inputan suara melalui program rekam suara yang telah dirancang. Lalu saat program mendeteksi inputan suara dari pengguna, program menyimpan data tersebut dalam file baru bentuk format wav dengan sample rate 16000. Setelah itu, data disiapkan untuk diproses melalui metode audio classification. Jika suara pengguna terdeteksi, program akan memprediksi kategori perintah yang sesuai berdasarkan model yang telah dilatih. Misalnya, jika suara pengguna mencocokkan pola "kipas mati," "kipas pejah," atau "kipas *off*," model akan memprediksi label "mati." Demikian pula, untuk pola "kipas tombol satu," "kipas tombol siji," atau "kipas tombol *one*," model akan memprediksi label "rendah." Pola "kipas tombol dua," "kipas tombol loro," atau "kipas tombol *two*" akan dikaitkan dengan label "sedang." Sementara pola "kipas tombol tiga," "kipas tombol telu," atau "kipas tombol *three*" akan diprediksi sebagai label "tinggi."

Dalam penelitian Tugas Akhir ini, dilakukan pengujian klasifikasi suara terhadap lima pengguna untuk mengevaluasi kinerja model secara lebih mendalam. Pengujian ini bertujuan untuk mengukur sejauh mana model mampu mengenali dan mengklasifikasikan perintah suara dari berbagai pengguna. Dengan menghadirkan variasi dalam data pengguna, diharapkan model dapat menangkap keragaman dalam pengucapan dan intonasi suara. Hasil dari pengujian ini menjadi kritikal dalam menilai apakah model memiliki generalisasi yang baik dan dapat digunakan secara luas.

### **3.6 Flowchart**

Gambaran alur kerja sistem dimulai dengan penggunaan mikrofon sebagai input untuk mendeteksi suara dalam tiga bahasa dan dua belas perintah berbeda. Suara yang terdeteksi mencakup instruksi seperti "kipas mati", "kipas tombol satu", "kipas tombol dua", "kipas tombol tiga", "kipas pejah", "kipas tombol siji", "kipas tombol loro", "kipas tombol telu", "kipas *off*", "kipas tombol *one*", "kipas tombol *two*", dan "kipas tombol *three*". Selanjutnya, suara yang terdeteksi akan diproses melalui langkah pengelompokan untuk

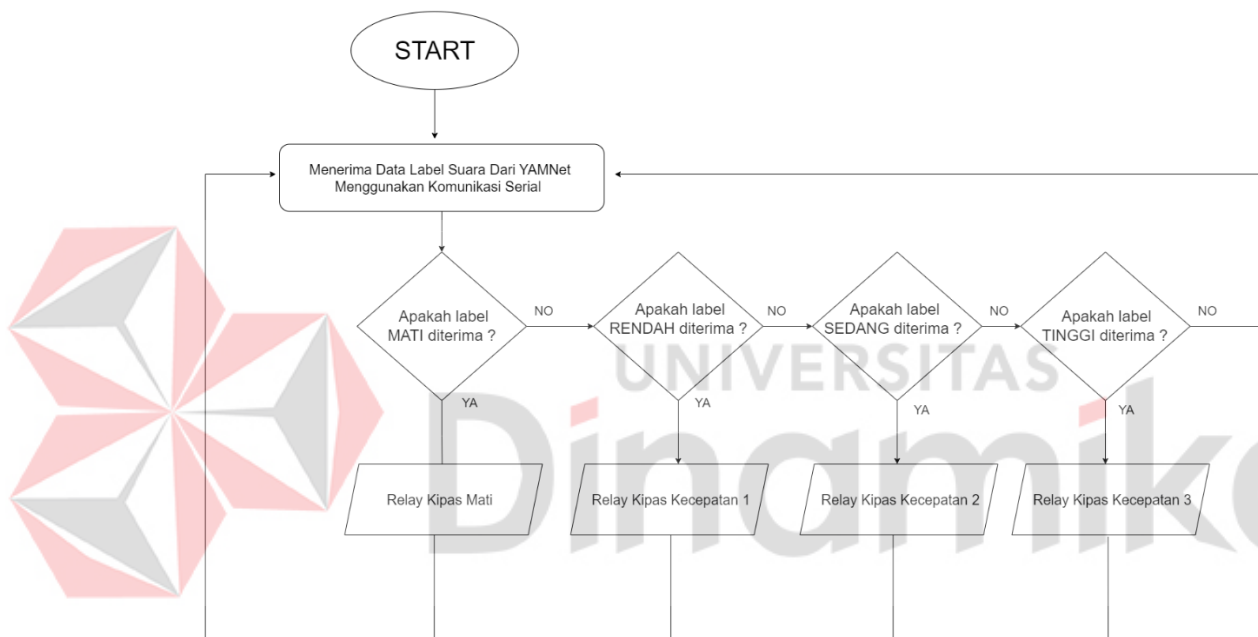
mengkategorikan ke dalam label "Mati", "Rendah", "Sedang", dan "Tinggi". Hasil dari pengenalan suara tersebut akan ditampilkan sebagai kalimat perintah yang terdeteksi bersama dengan label kategorinya.



Gambar 3.21 Flowchart Python

Setelah proses pengolahan data pada program Python menggunakan metode klasifikasi audio, data hasil pengenalan tersebut akan dikirim ke Arduino Uno melalui komunikasi serial. Pada Arduino Uno, data yang diterima akan dilakukan pengecekan kondisi untuk memastikan label yang dihasilkan sesuai dengan perintah pengguna. Jika label yang diterima adalah "Mati", maka relay 1 pada Arduino Uno akan diaktifkan, menandakan bahwa kipas angin dalam keadaan mati. Jika label yang diterima adalah "Rendah", relay 2 akan diaktifkan, menandakan bahwa kipas angin beroperasi pada tingkat satu atau rendah. Untuk label "Sedang", relay 3 akan diaktifkan, menunjukkan bahwa kipas angin berada pada

tingkat dua atau sedang. Terakhir, jika label yang diterima adalah "Tinggi", relay 4 akan diaktifkan, menandakan bahwa kipas angin beroperasi pada tingkat tiga atau tinggi. Selanjutnya, sistem akan kembali ke aplikasi rekaman suara untuk memulai siklus pengenalan suara berikutnya. Dengan pengintegrasian yang tepat antara alur kerja Python dan Arduino Uno, sistem dapat memberikan respons yang efektif sesuai dengan perintah suara yang diberikan oleh pengguna.



**Gambar 3.22** Flowchart Arduino Uno

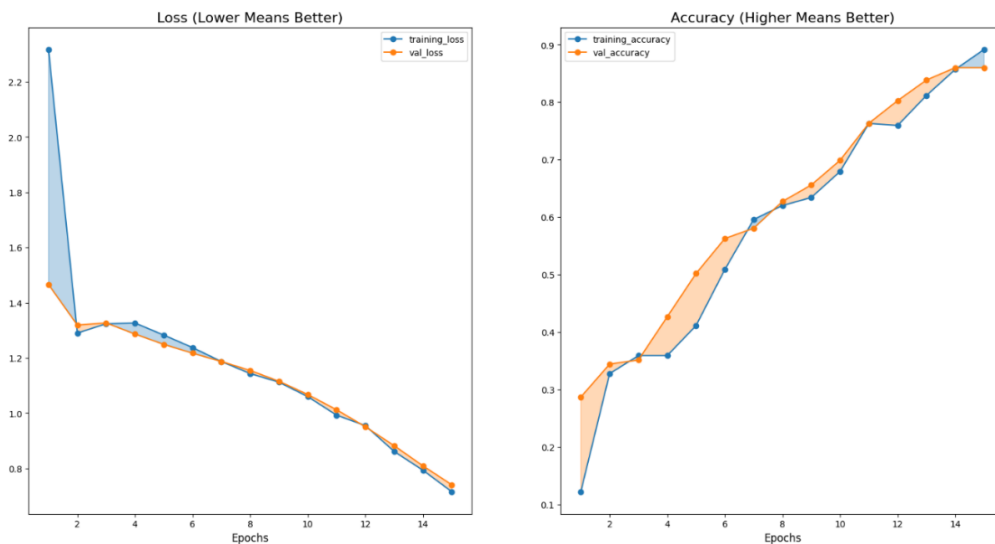


## BAB IV

### HASIL DAN PEMBAHASAN

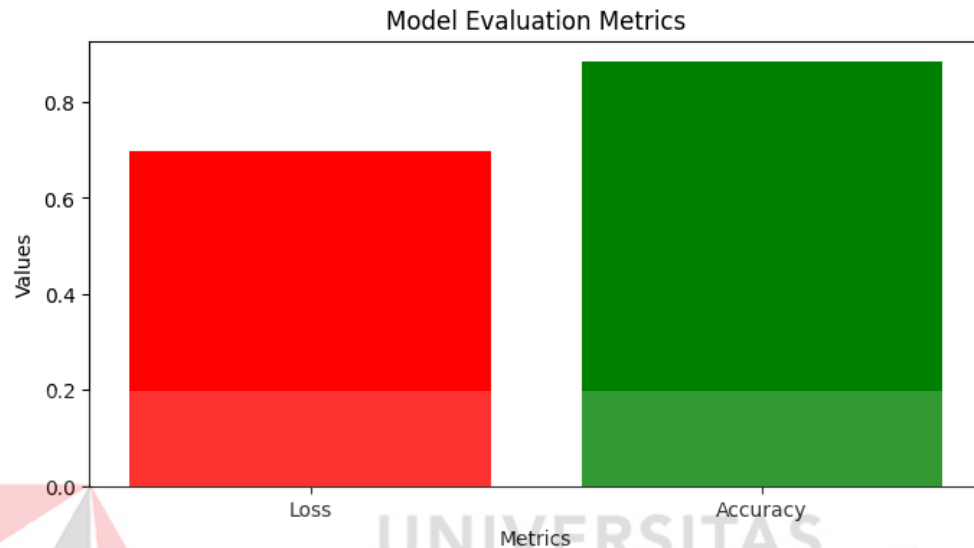
#### 4.1 Hasil Proses Training Model Pertama Dengan 12 Epoch

Pada hasil percobaan, model yang menggunakan *epoch* sebanyak 12 menunjukkan peningkatan kinerja yang signifikan. Akurasi pada data *training* mencapai 89%, sedangkan nilai *loss* hanya sebesar 0.71, sebagaimana tergambar dengan jelas pada Gambar 4.1. Tingginya akurasi ini memberikan indikasi positif terhadap kemampuan model dalam menggeneralisasi pola dari data training. Dengan nilai *loss* yang rendah, dapat diartikan bahwa model mampu meminimalkan kesalahan prediksi selama proses pelatihan. Peningkatan akurasi ini mencerminkan keberhasilan model dalam memahami pola – pola kompleks dalam data latih. Tingginya akurasi pada data training memberikan keyakinan tambahan terhadap kemampuan model dalam melakukan klasifikasi. Semakin tinggi akurasi, semakin baik kinerja model tersebut dalam mengidentifikasi pola – pola yang mendasari data *train*. Akurasi yang mencapai 89% menandakan bahwa model mampu memprediksi akurasi tersebut dengan nilai *loss* yang rendah menciptakan kombinasi ideal antara akurasi dan ketepatan prediksi model.



Gambar 4.1 Akurasi dan *Loss* Model Pertama

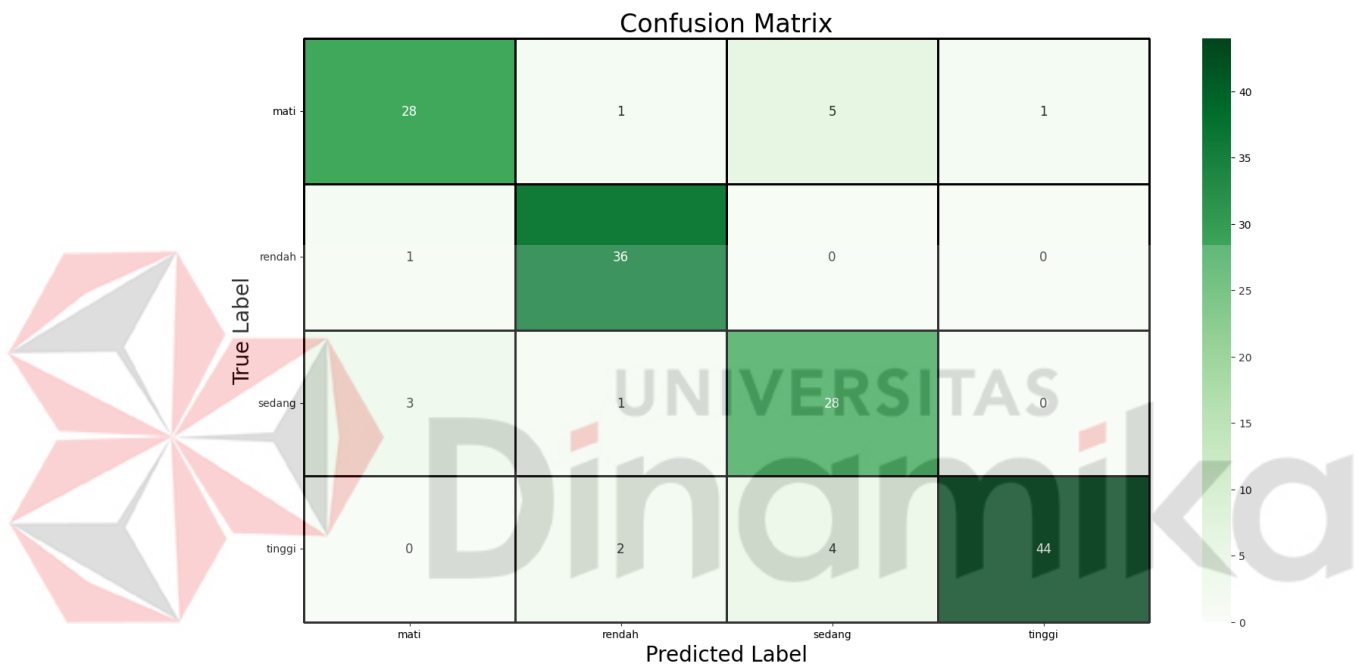
Namun, dalam mengevaluasi kinerja model, penting juga untuk memperhatikan hasil pada data evaluasi. Evaluasi pada dataset validasi menghasilkan akurasi sebesar 88% dengan nilai *loss* 0.72 (Gambar 4.2), yang tetap mencerminkan kinerja yang baik walaupun dengan sedikit penurunan dibandingkan dengan data training.



Gambar 4.2 Evaluasi Akurasi dan *Loss* Model Pertama

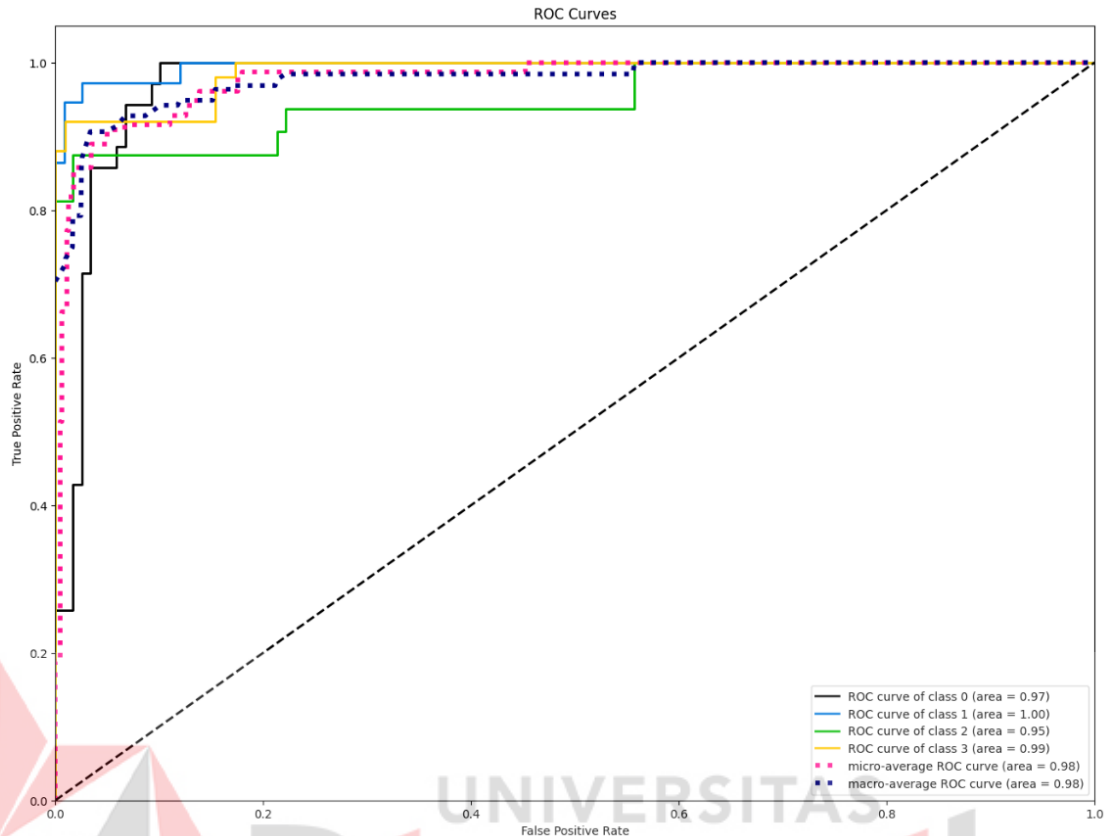
Berdasarkan analisis *confusion matrix*, hasil dari pelatihan model pertama menunjukkan adanya ketidaksesuaian pada hasil prediksi untuk semua kelas (Gambar 4.3). meskipun akurasi umumnya tinggi dengan nilai 89%, tampaknya terdapat kesalahan dalam memprediksi kelas – kelas tertentu. Hal ini dapat merujuk pada adanya *false positives* dan *false negatives* yang dapat mempengaruhi kualitas prediksi model. *Confusion matrix* memberikan gambaran rinci tentang sejauh mana model berhasil membedakan setiap kelas, dan hasil yang tidak sesuai menunjukkan tantangan yang perlu diatasi. Pentingnya pemahaman terhadap hasil *confusion matrix* menjadi jelas bahwa dalam konteks evaluasi model klasifikasi. Meskipun akurasi secara umum memberikan gambaran tentang seberapa baik model dapat memprediksi kelas secara keseluruhan, namun informasi tambahan dari *confusion matrix* memberikan wawasan lebih mendalam tentang kinerja model pada setiap kelas khususnya. Oleh karena itu, keberhasilan model tidak hanya dapat diukur dari tingginya akurasi global, tetapi harus mempertimbangkan kemampuan model dalam memprediksi setiap kelas dengan tepat. Dengan menyoroti kesalahan prediksi pada *confusion matrix*, penelitian lebih lanjut dan perbaikan model dapat dilakukan untuk

meningkatkan performa khususnya pada kelas – kelas yang mendemonstrasikan ketidaksesuaian. Analisis lebih lanjut terhadap factor-faktor yang mempengaruhi hasil prediksi, seperti ketidakseimbangan data atau kompleksitas pola kelas tertentu, dapat membantu mengidentifikasi strategi yang diperlukan untuk meningkatkan akurasi model pada kasus – kasus tertentu.



Gambar 4.3 *Confusion Matrix*

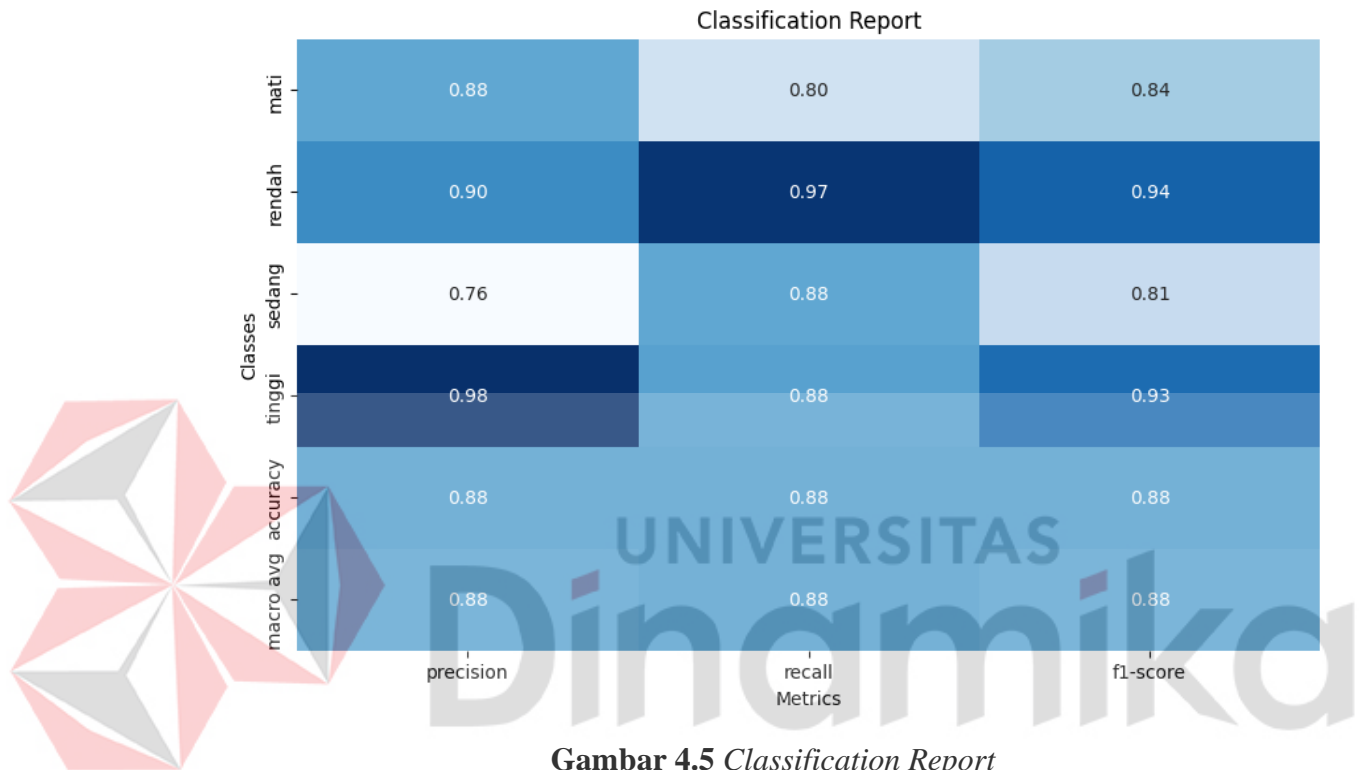
Dan pada ROC *curve* menunjukkan bahwa kinerja model yang telah di latih menghasilkan yang cukup baik. ROC *curve* pada Gambar 4.4 menampilkan bahwa beberapa class mencapai nilai 0.9, menunjukkan kemampuan model untuk membedakan dengan baik antara kelas – kelas tersebut. Selain itu, kelas 1 atau kelas rendah bahkan mencapai nilai 1.00 atau 100%, mengindikasikan tingkat keberhasilan yang sangat tinggi dalam mengklasifikasikan observasi positif untuk *class* tersebut. Fenomena ini dapat dianggap sebagai indikator kehandalan model dalam mengenali dan memprediksi suatu kelas secara akurat.



Gambar 4.4 ROC Curve Model Pertama

Selanjutnya, analisis hasil pada *classification report* memberikan wawasan lebih rinci terkait performa model pada masing – masing class. Meskipun terdapat *class – class* yang mencapai akurasi tinggi, seperti *class* “tinggi” dengan nilai 98% seperti yang terlihat pada Gambar 4.5, tetapi terdapat juga kelas yang menunjukkan performa yang lebih rendah, seperti kelas “sedang” dengan akurasi 76%. perbedaan signifikan ini memberikan wawasan yang lebih komprehensif tentang kemampuan model dalam mengklasifikasikan masing – masing kelas. Kesuksesan model dalam mengidentifikasi *class* “tinggi” menunjukkan kemampuan yang baik dalam kasus tersebut, sementara kinerja yang lebih rendah pada kelas “sedang” menandakan potensi area perbaikan yang perlu diperhatikan. Dengan demikian, evaluasi pada setiap kelas memberikan pemahaman yang lebih holistic tentang seberapa baik model dapat menangani variasi antar *class*. Hasil ini dapat menjadi panduan penting dalam mengidentifikasi aspek – aspek tertentu yang perlu dioptimalkan atau dioptimalkan Kembali untuk meningkatkan kinerja klasifikasi secara keseluruhan. Dengan menyoroti

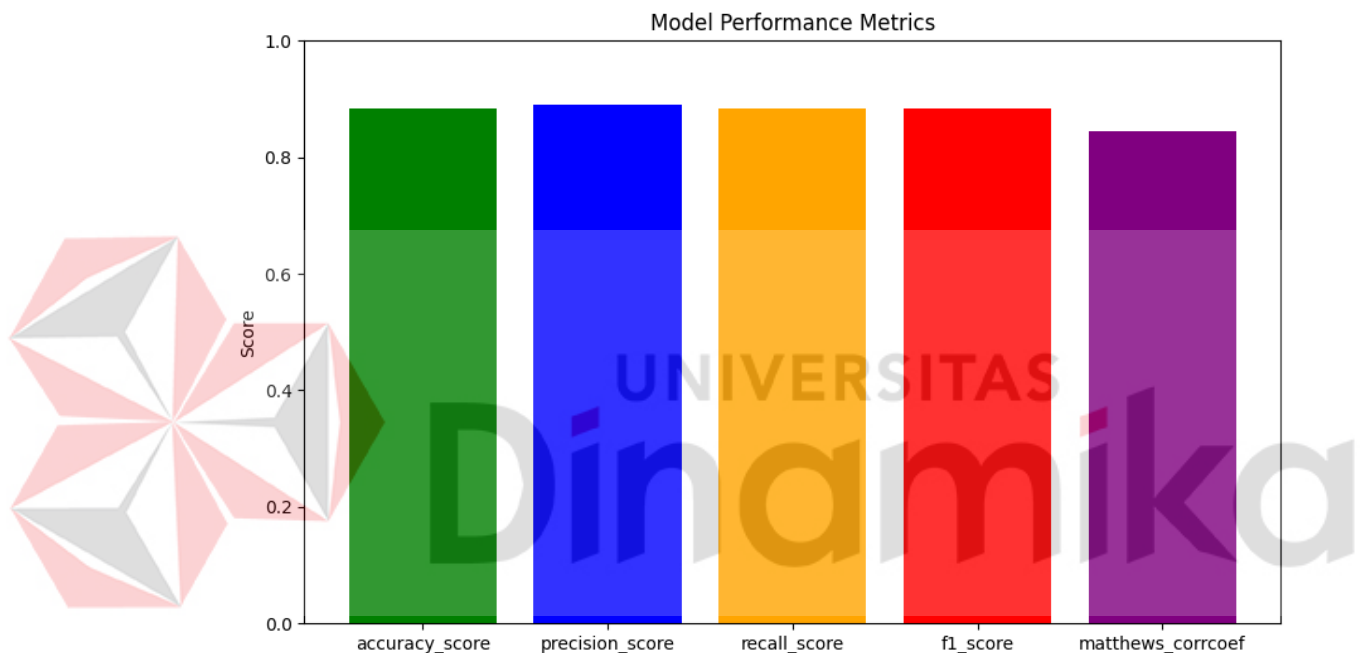
perbedaan kinerja antar class, analisis classification report memberikan petunjuk berharga untuk pengembangan model lebih lanjut dan pengoptimalan parameter agar dapat mengatasi ketidakseimbangan kinerja antar *class*, memastikan bahwa model dapat memberikan prediksi yang akurat dan konsisten di semua kategori yang dihadapi.



Gambar 4.5 Classification Report

Dalam memeriksa dari model pertama menunjukkan bahwa dalam evaluasi kinerjanya, terdapat sedikit perbedaan antara skor akurasi, presisi, *recall*, dan *f1 – score*, dengan nilai Matthews yang terlihat pada Gambar 4.6. model berhasil mencapai akurasi tinggi secara keseluruhan, tetapi perhatian khusus diperlukan pada *class* “sedang” yang menunjukkan akurasi rendah sebesar 76% pada Gambar 4.5. evaluasi kelas lain menunjukkan hasil yang lebih memuaskan, seperti kelas “tinggi” dengan akurasi mencapai 0.98. analisis lebih lanjut pada *performance metric* mengungkapkan kesesuaian yang era tantara skor akurasi, *recall*, dan *f1 – score*, menunjukkan konsistensi model dalam melakukan prediksi. Perbandingan dengan nilai Matthews memberikan gambaran tambahan terkait sejauh mana model mampu mengatasi ketidakseimbangan *class*. Selain itu, penilaian kinerja model melibatkan pemeriksaan ROC curve yang menunjukkan beberapa *class*

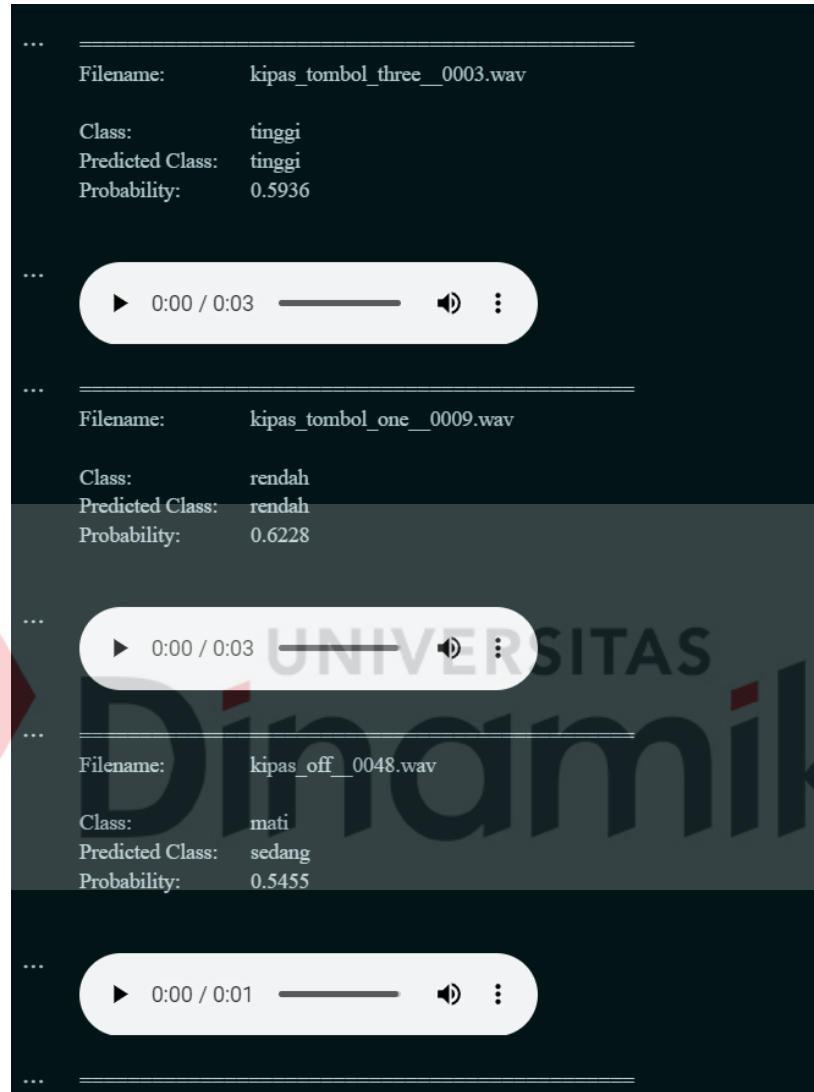
dengan nilai mencapai 0.9, dan satu kelas mencapai nilai sempurna, yaitu 1.00 atau 100% (Gambar 4.4). *confusion matrix* memberikan gambaran lebih rinci tentang prediksi yang tidak sesuai pada berbagai class (Gambar 4.3). dalam upaya untuk memahami ketepatan model dalam mengklasifikasikan setiap kelas, *classification report* memberikan informasi terperinci mengenai presisi, *recall* dan *f1 – score* untuk setiap kelas. Perbandingan antara *class – class* ini memberikan wawasan yang berguna terkait dengan kekuatan dan kelemahan model dalam menangani masing – masing kategori.



Gambar 4.6 Model Performance Metrics Pertama

Uji kinerja pada dataset *test* merupakan tahap kritis dalam menilai sejauh mana model dapat diaplikasikan dalam situasi dunia nyata, sebagaimana tergambar pada Gambar 4.7. hasil dari uji ini memberikan pemahaman praktis tentang kelayakan dan keandalan model diluar lingkup pelatihan. Temuan ini memberikan konteks tambahan yang sangat berharga dalam mengevaluasi efektivitas model dalam menghadapi variasi dan kompleksitas situasi dunia nyata. Secara keseluruhan, penelitian ini menyoroti kebutuhan mendesak untuk meningkatkan kinerja pada class “sedang”, yang tercatat memiliki akurasi yang lebih rendah. Evaluasi lebih lanjut diperlukan untuk menganalisis faktor – faktor yang mungkin mempengaruhi kinerja model dalam class tersebut. Penelitian ini secara jelas menunjukkan

bahwa untuk menghadapi tantangan dunia nyata, model perlu disesuaikan dan diperbaiki agar dapat mengatasi Dinamika yang kompleks dan bervariasi.



Gambar 4.7 Pengujian Kinerja Model Menggunakan Model Pertama

Dalam proses pengujian klasifikasi audio menggunakan model pertama yang dilakukan sebanyak 40 kali, dimana pengujian tersebut mencakup seluruh perintah yang ada dan pengujian setiap masing – masing perintah dilakukan sebanyak 10 kali untuk mengatur kipas angin yang diuji untuk memastikan respon model yang akurat dan konsisten terhadap setiap perintah level dalam kecepatan kipas angin. Pengujian ini bertujuan untuk

mengevaluasi kemampuan model dalam memahami konteks spesifik dari setiap perintah dan respon yang sesuai.

Tabel 4.1 Pengujian Dataset *Subset Test* Menggunakan Model Pertama

BAHASA	STATUS KIPAS	AKURASI	
		BENAR	SALAH
Bahasa Indonesia	0	2	2
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	1	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	2	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	3	4	0
Bahasa Jawa		3	0
Bahasa Inggris		2	1
Jumlah		37	3
Akurasi		92.5%	7.5%

Pengujian pada subset test dataset dengan model yang telah dilatih menggunakan 12 *epoch* menunjukkan tingkat akurasi sebesar 92.5%, mengindikasikan kemampuan yang baik dalam melakukan klasifikasi audio. Dengan hasil kinerja tersebut, model ini mampu dengan efektif membedakan dan mengklasifikasikan data audio sesuai dengan tugas yang dapat dilaksanakan. Keberhasilan ini mencerminkan tingkat performa yang tinggi dan memberikan keyakinan bahwa model tersebut dapat diandalkan dalam proses klasifikasi audio. Hasil akurasi tersebut menunjukkan bahwa model telah mampu memahami dan menangkap audio



dengan baik dan menegaskan bahwa pendekatan yang diambil dalam pelatihan dan arsitektur model memberikan hasil yang memuaskan.

Tabel 4.2 Pengujian Pengguna Suara Yang Terdapat Dalam Model Dataset Pertama

NAMA PENGGUNA A	BAHASA	STATU S KIPAS	AKURASI		KETERANGAN	
			BENAR	SALAH		
Pegguna Suara Yang Terdapat Dalam Dataset	Kipas Mati	0	6	4	Akurasi benar = 43,3% Akurasi salah = 56,7%	
	Kipas Pejah		3	7		
	Kipas Off		4	6		
	Jumlah			13	17	
	Kipas Tombol Satu	1	4	6	Akurasi benar = 40% Akurasi salah = 60%	
	Kipas Tombol Siji		5	5		
	Kipas Tombol One		3	7		
	Jumlah			12	18	
	Kipas Tombol Dua	2	5	5	Akurasi benar = 60% Akurasi salah = 40%	
	Kipas Tombol Loro		6	4		
	Kipas Tombol Two		7	3		
	JUMLAH			18	12	
	Kipas Tombol Tiga	3	6	4	Akurasi benar = 43,3% Akurasi salah = 56,7%	
Kipas Tombol Telu	3		7			
Kipas Tombol Three	4		6			
Jumlah			13	17		

Hasil pengujian klasifikasi suara pengguna yang terdapat dalam dataset merupakan suara pengguna tersebut juga berada dalam dataset yang telah melewati proses training dengan 12 *epoch* menunjukkan variabilitas dalam akurasi prediksi untuk setiap kategori status. Untuk kategori status 0, akurasi benar sebesar 43,3%, yang berarti model berhasil mengidentifikasi status 0 dengan benar pada proporsi tersebut. Sementara itu, terdapat 56,7% prediksi yang salah untuk status 0. Status 1 memiliki akurasi benar sebesar 40%,

dan sebaliknya, memiliki akurasi salah sebesar 60%. Status 2 menunjukkan performa lebih baik dengan akurasi benar sebesar 60%, yang menandakan kemampuan model dalam mengklasifikasikan status 2 secara tepat. Meskipun status 3 memiliki akurasi benar yang sama dengan status 0, yaitu 43,3%, tingkat akurasi salahnya mencapai 56,7%.

Tabel 4.3 Pengujian Suara Pengguna Yang Tidak Terdapat Dalam Model Dataset Pertama

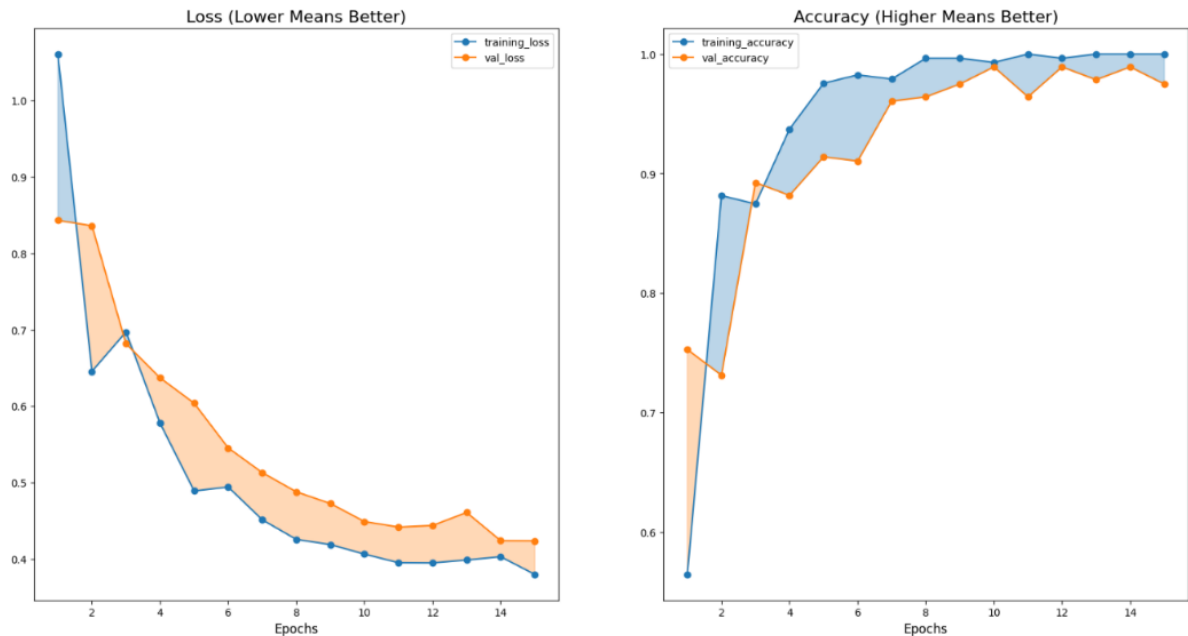
NAMA PENGGUNA	BAHASA	STATUS KIPAS	AKURASI		KETERANGAN	
			BENAR	SALAH		
Pengguna Suara Yang Tidak Terdapat Dalam Dataset	Kipas Mati	0	2	8	Akurasi benar = 10% Akurasi salah = 90%	
	Kipas Pejah		1	9		
	Kipas Off		0	10		
	Jumlah			3	27	
	Kipas Tombol Satu	1	4	6	Akurasi benar = 30% Akurasi salah = 70%	
	Kipas Tombol Siji		4	6		
	Kipas Tombol One		1	9		
	Jumlah			9	21	
	Kipas Tombol Dua	2	4	6	Akurasi benar = 56,7% Akurasi salah = 43,3%	
	Kipas Tombol Loro		6	4		
	Kipas Tombol Two		7	3		
	Jumlah			17	13	
	Kipas Tombol Tiga	3	3	7	Akurasi benar = 20% Akurasi salah = 80%	
	Kipas Tombol Telu		1	9		
	Kipas Tombol Three		2	8		
	JUMLAH			6	24	

Berdasarkan hasil pengujian klasifikasi suara pengguna yang tidak terdapat pada dataset merupakan suara pengguna yang diinput melalui mic tidak tersedia di dalam dataset pada saat training dengan 12 epoch yang dapat dilihat bahwa tingkat akurasi klasifikasi varian status 0 menunjukkan hasil rendah, dengan hanya 10% akurasi benar dan 90%

akurasi salah. Sebaliknya, status 1 memiliki akurasi benar sebesar 30%, yang meskipun meningkat dibandingkan dengan status 0, masih menunjukkan tingkat akurasi yang relatif rendah, dengan akurasi salah mencapai 70%. Peningkatan yang lebih signifikan terlihat pada status 2, dengan akurasi benar mencapai 56,7% dan akurasi salah sebesar 43,3%. Namun, meskipun mencapai tingkat akurasi yang lebih baik, status 2 juga memiliki ruang untuk peningkatan. Sementara itu, status 3 menunjukkan akurasi benar sebesar 20%, yang menandakan tingkat klasifikasi yang rendah, dengan akurasi salah mencapai 80%.

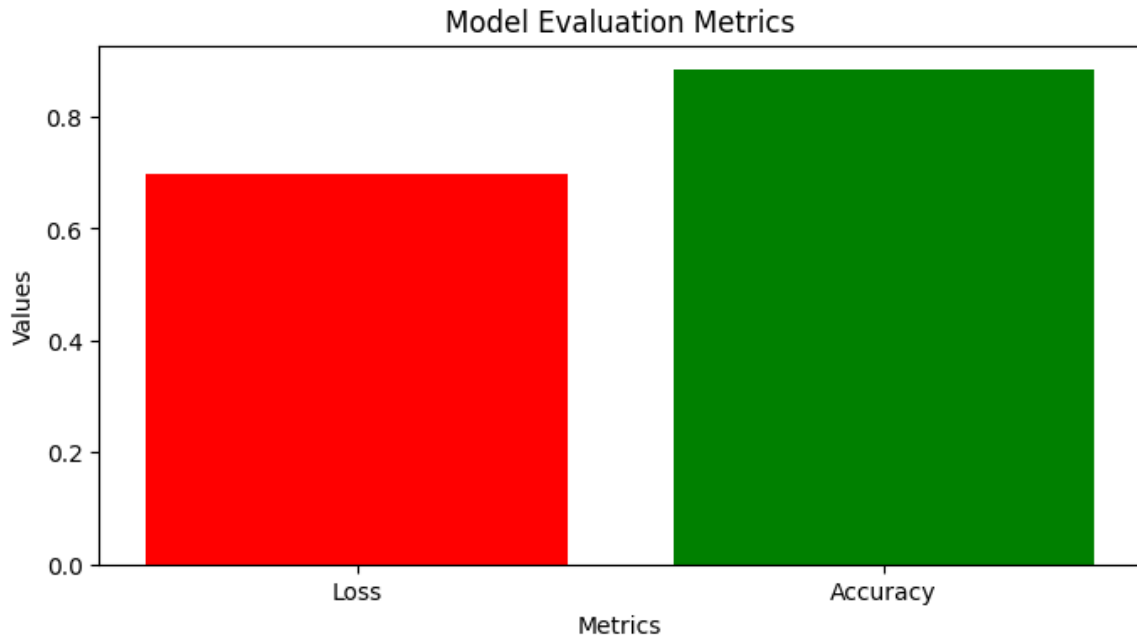
#### **4.2 Hasil Proses *Training Model* Kedua dengan 15 Epoch**

Hasil proses pelatihan data kedua menunjukkan hasil yang signifikan, dengan akurasi akhir mencapai 1.00 atau 100%, dan nilai *loss* yang menurun menjadi 0.39. peningkatan performa ini menandai kemajuan yang mencolok dibandingkan dengan model pelatihan pertama. Dengan akurasi penuh, model ini mampu memberikan prediksi yang sempurna sesuai dengan data pelatihan. Hasil ini memberikan indikasi bahwa model telah berhasil menyesuaikan diri dengan pola – pola kompleks dalam data dan dapat memprediksi kelas dengan presisi tinggi. Nilai *loss* yang rendah juga menunjukkan bahwa model telah efisien dalam mengoptimalkan bobotnya selama proses pelatihan. Peningkatan ini dapat disebabkan oleh berbagai faktor, seperti penyesuaian parameter, penambahan data pelatihan, atau penggunaan Teknik pelatihan yang lebih canggih. Meskipun demikian, perlu dicatat bahwa akurasi maksimal ini juga dapat menandakan overfitting, dimana model mungkin terlalu mempertimbangkan noise atau detail kecil dalam data pelatihan yang mungkin tidak mencerminkan pola umum. Oleh karena itu, validasi pada dataset uji dan analisis lebih lanjut perlu dilakukan untuk memastikan keandalan model dalam situasi dunia nyata. Peningkatan kinerja pada model kedua ini memberikan landasan yang solid untuk peningkatan lebih lanjut, dan pemahaman mendalam terhadap faktor – faktor yang menyebabkan perbaikan ini dapat membimbing pengembangan model yang lebih baik di masa depan.



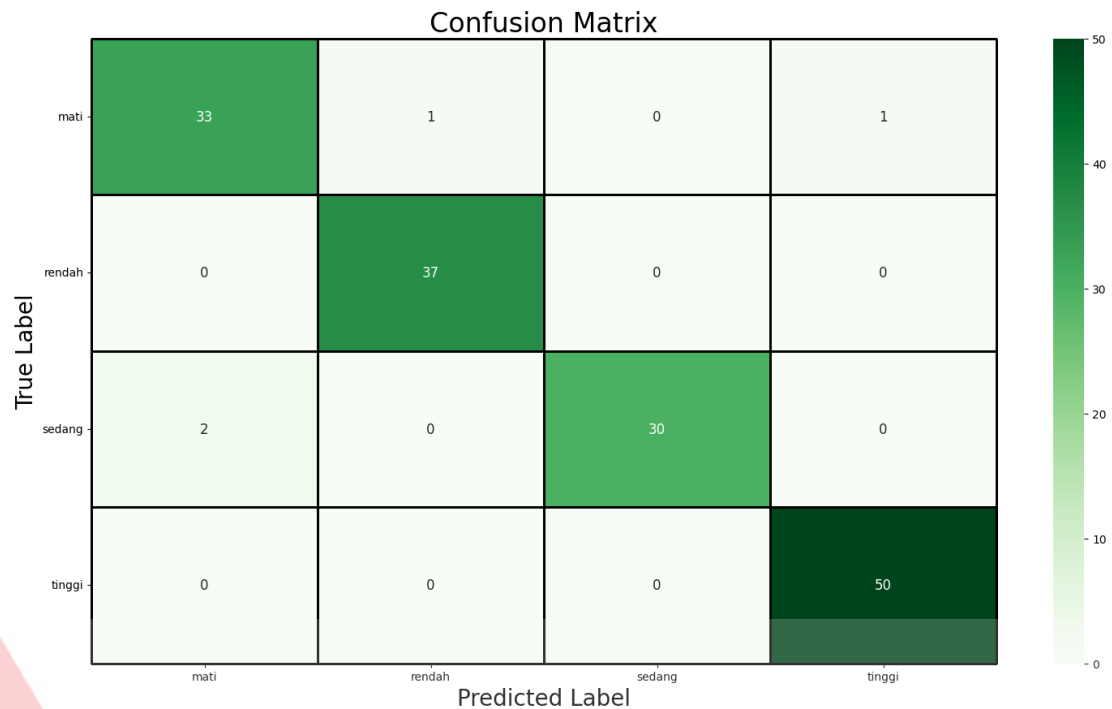
Gambar 4.8 Akurasi dan *Loss* Model Kedua

Hasil evaluasi model pada tahap pelatihan kedua menunjukkan peningkatan kinerja yang signifikan, dengan akurasi mencapai angka diatas 0.90 dan *loss* sebesar 0.69. peningkatan ini memberikan indikasi positif bahwa model mengalami pembelajaran yang lebih efektif dan dapat menghasilkan prediksi yang lebih akurat setelah iterasi kedua. Dengan akurasi yang mencapai tingkat tersebut, dapat disimpulkan bahwa model telah berhasil mengatasi beberapa kelemahan atau ketidaksempurnaan yang mungkin ada pada iterasi sebelumnya. Selain itu, penurunan nilai *loss* sebesar 0.69 menunjukkan bahwa model mampu mengoptimalkan proses pembelajarannya dan meningkatkan kemampuan generalisasinya terhadap data baru. Hasil ini memberikan konfirmasi bahwa proses pelatihan kedua membawa dampak positif terhadap kualitas prediksi model. peningkatan kinerja ini dapat memiliki implikasi penting dalam konteks penerapan praktis, mengingat model sekarang mampu memberikan hasil dengan tingkat ketepatan yang lebih tinggi. Oleh karena itu, hasil evaluasi ini memberikan dukungan kuat terhadap Keputusan untuk melibatkan pelatihan model lebih lanjut, yang kemudian dapat memberikan manfaat nyata dalam penerapannya pada skenario dunia nyata.



Gambar 4.9 Evaluasi Model Kedua

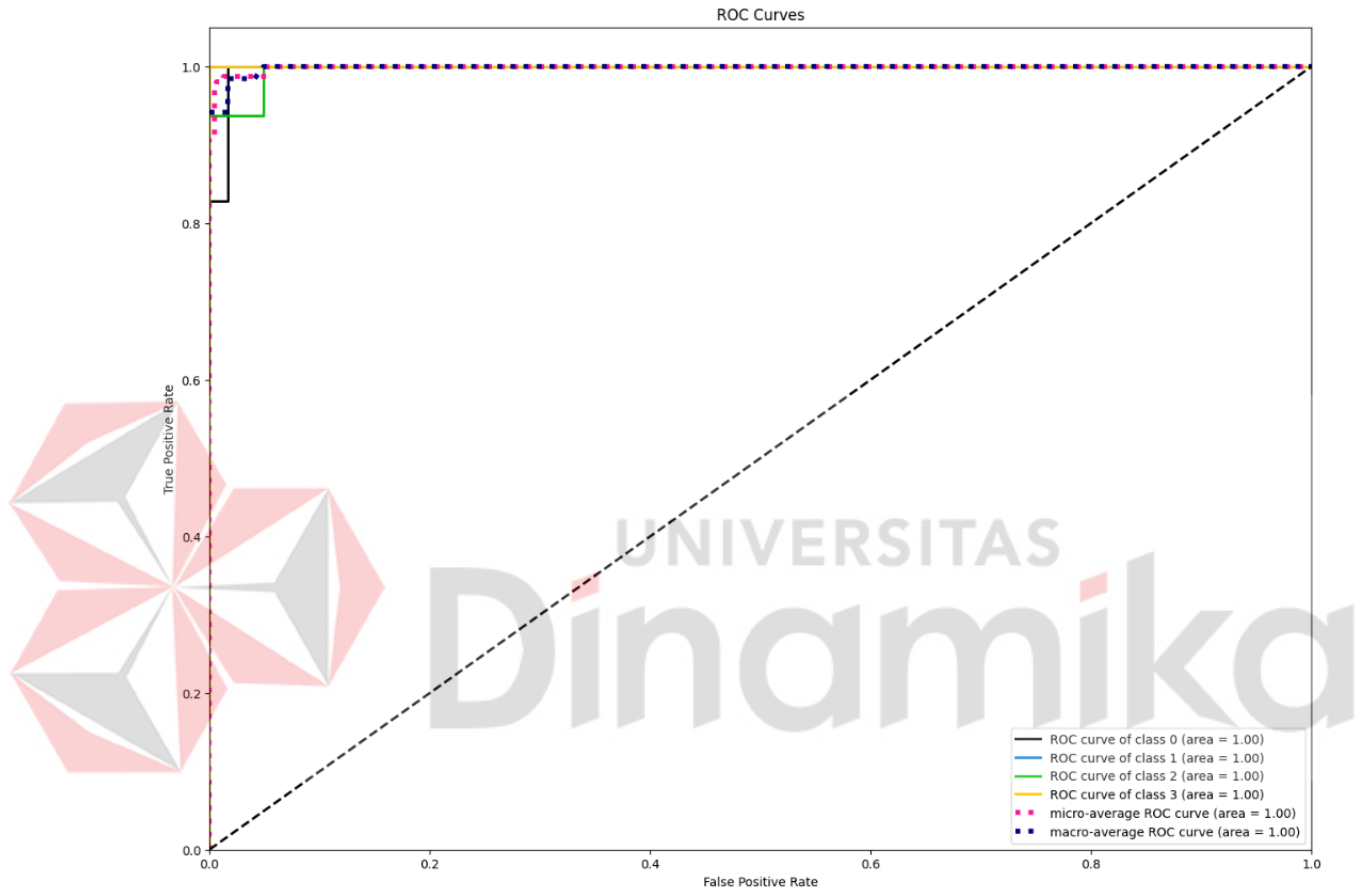
Hasil analisis prediksi menggunakan *confusion matrix* pada model pelatihan kedua yang terlihat pada Gambar 4.10 menunjukkan peningkatan yang signifikan dibandingkan dengan model sebelumnya, meskipun masih terdapat beberapa kesalahan dalam memprediksi beberapa kelas. Peningkatan tersebut terutama terlihat pada kelas rendah dan tinggi, dimana model mampu memberikan hasil prediksi yang lebih akurat (Gambar 4.8). meskipun begitu, perlu dicatat bahwa hasil prediksi yang baik pada kelas tertentu tidak menjamin keberhasilan model ketika dihadapkan pada situasi dunia nyata, terutama ketika menghadapi klasifikasi suara pengguna yang tidak terdapat dalam dataset pelatihan ini. Penting untuk menyadari bahwa keberhasilan model dalam mengklasifikasikan suara pengguna yang belum pernah ditemui sebelumnya dapat menjadi tantangan yang signifikan. Meskipun kelas rendah dan tinggi telah menghasilkan prediksi yang baik pada dataset pelatihan, tidak dapat dipastikan bahwa model akan sama efektifnya ketika dihadapkan pada variasi dan kompleksitas suara pengguna yang berbeda. Oleh karena itu, diperlukan strategi yang cermat dalam mengadaptasi model untuk mengenali pola suara yang tidak terwakili dalam dataset pelatihan. Meskipun masih terdapat sedikit kesalahan dalam prediksi kelas, hasil yang lebih baik pada model kedua memberikan arah positif untuk pengembangan lebih lanjut.



Gambar 4.10 *Confusion Matrix* Model Kedua

Terdapat temuan dari *ROC curve* yang menunjukkan bahwa seluruh kelas pada model berhasil mencapai akurasi sempurna sebesar 1.00. temuan ini secara khusus terlihat melalui visualisasi grafik *ROC curve* yang menggambarkan nilai *True Positive Rate* (TPR) versus *False Positive Rate* (FPR) untuk setiap kelas. Pengamatan ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antara kelas – kelas yang berbeda, dimana semua nilai area di bawah kurva (*Area Under Curve/ AUC*) mencapai nilai maksimal yaitu 1.00. hasil ini mencerminkan tingkat ketepatan model dalam mengklasifikasikan instance positif dan negatif untuk setiap kelas, tanpa kesalahan prediksi apapun. Meskipun keberhasilan ini sangat menggembirakan, penting untuk tetap melihatnya dalam konteks lebih luas, mempertimbangkan potensi efek samping seperti *overfitting* pada data pelatihan yang mungkin memengaruhi generalisasi model pada dataset baru. Oleh karena itu, sementara hasil *ROC curve* memberikan indikasi bahwa model pada dataset baru. Oleh karena itu, sementara hasil *ROC curve* memberikan indikasi bahwa model memiliki performa yang sangat baik, diperlukan analisis lebih lanjut untuk memastikan keandalan model di berbagai kondisi dan situasi dunia nyata. Selain itu,

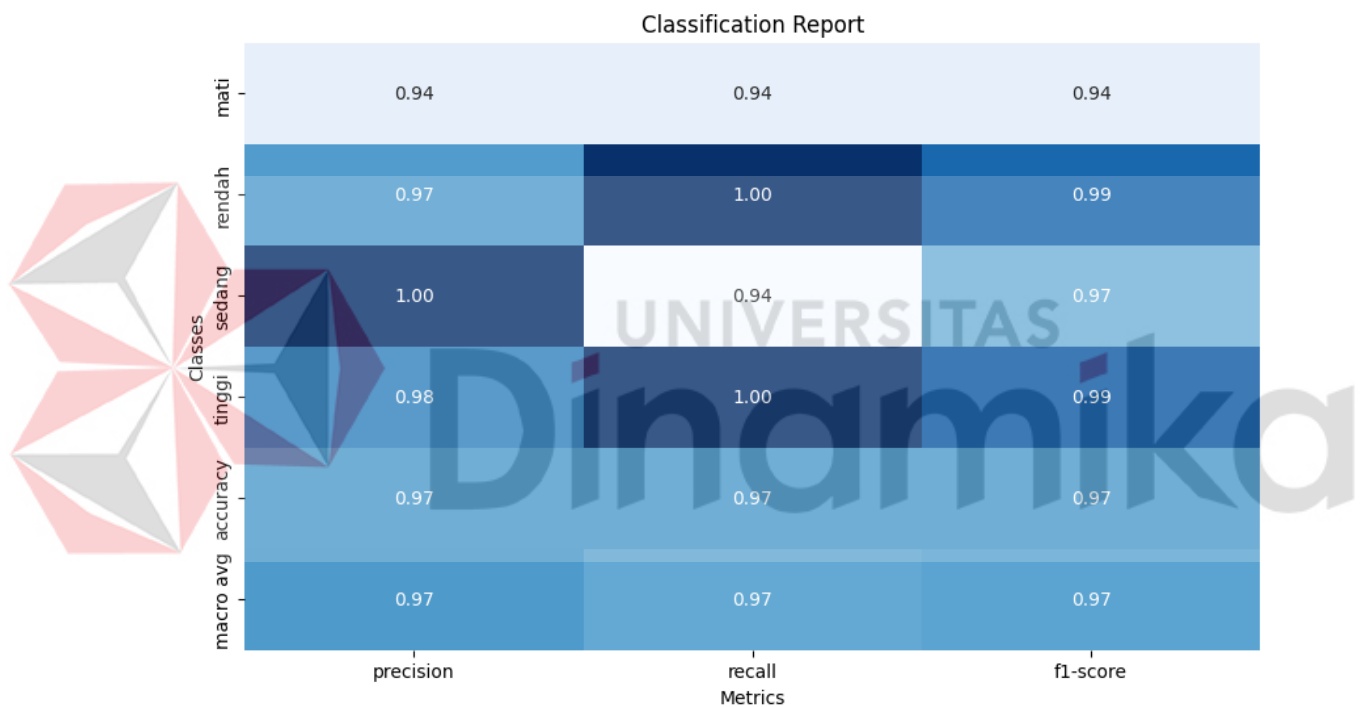
evaluasi model juga harus mempertimbangkan metrik kinerja lainnya seperti *precision*, *recall*, dan *F1 – score* untuk mendapatkan gambaran komprehensif tentang performa model di seluruh kelas. Meskipun ROC curve memberikan gambaran positif, interpretasi yang *holistic* melalui berbagai metrik membantu menginformasikan langkah – langkah perbaikan dan peningkatan yang mungkin diperlukan dalam pengembangan model.



Gambar 4.11 ROC Curve Model Kedua

Selanjutnya, hasil *classification report* memberikan gambaran yang mendetail terkait pengukuran akurasi menggunakan precision pada berbagai kelas. Ditemukan bahwa kelas “sedang” mencapai akurasi 1.00, sementara kelas lainnya memiliki akurasi sekitar 0.90, menunjukkan kinerja yang masih cukup baik. Akurasi keseluruhan mencapai 0.97, dengan rata – rata mako sebesar 0.97 (Gambar 4.12). pada pengukuran *recall*, terdapat dua kelas yaitu “rendah” dan “tinggi” yang mencapai akurasi 1.00, sementara dua kelas lainnya mencapai akurasi 0.94. akurasi keseluruhan tetap tinggi, yaitu 0.97, dengan rata – rata

makro sebesar 0.97. dalam pengukuran *f1-score*, beberapa kelas mengalami sedikit penurunan, tetapi masih tergolong baik. Meskipun ada variasi akurasi dari setiap pengukuran kelas, namun kelas “mati” menunjukkan hasil akurasi yang konsisten pada *precision*, *recall*, dan *f1-score*. Temuan ini mengindikasikan stabilitas kinerja kelas “mati” sepanjang pengukuran, menyoroti keandalan model terhadap kelas tersebut. Secara keseluruhan, hasil ini menggambarkan bahwa model tetap memiliki kinerja yang baik dengan akurasi tinggi dan rata – rata makro yang konsisten, namun perlu perhatian khusus terhadap beberapa kelas yang mengalami penurunan akurasi pada beberapa pengukuran.

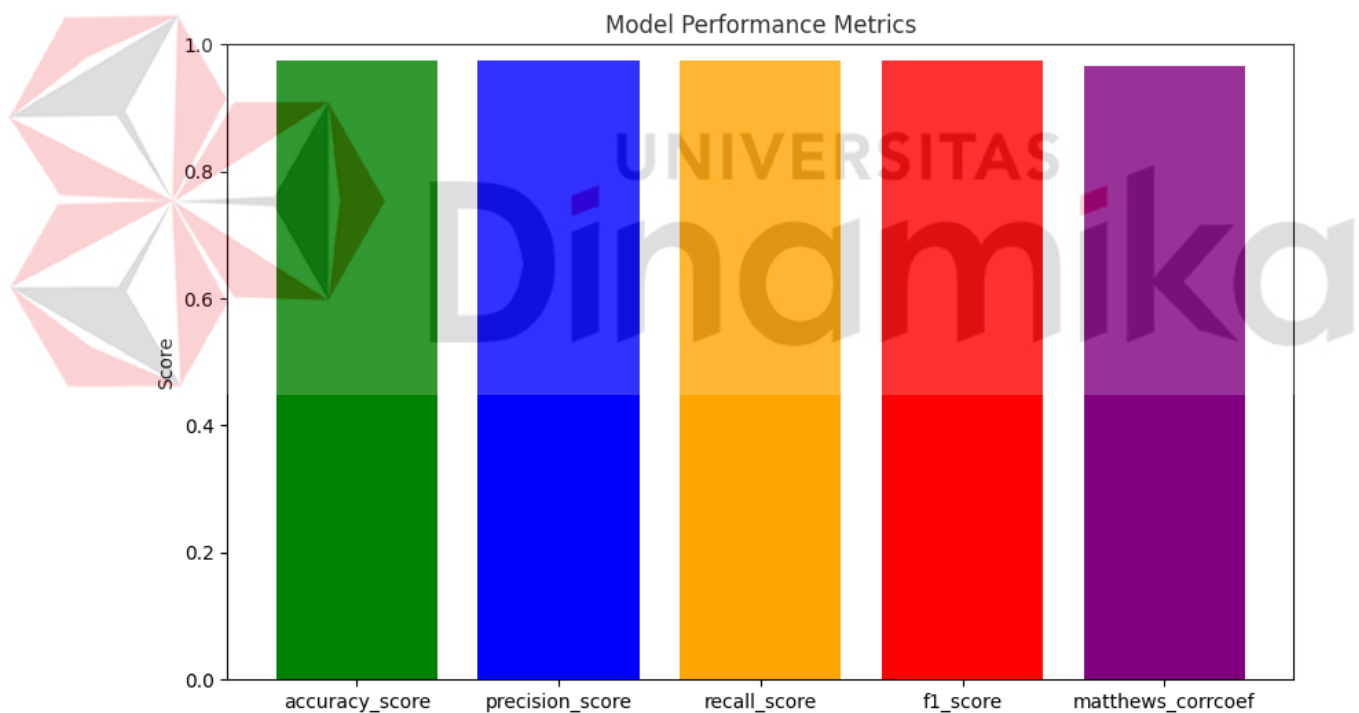


Gambar 4.12 *Classification Report* Model Kedua

Selanjutnya, analisis hasil *model performance metric* menunjukkan keseragaman dalam hasil semua pengukuran, namun terdapat perbedaan yang mencolok pada pengukuran Matthews, tergambar dengan jelas pada Gambar 4.13. Meskipun nilai akurasi, presisi, *recall*, dan *f1 – score* menunjukkan kesamaan, penekanan pada metrik Matthews menjadi poin kunci dalam evaluasi kinerja keseluruhan. Perbedaan pada pengukuran ini mengindikasikan bahwa model mungkin memiliki kecenderungan tertentu atau sensitivitas



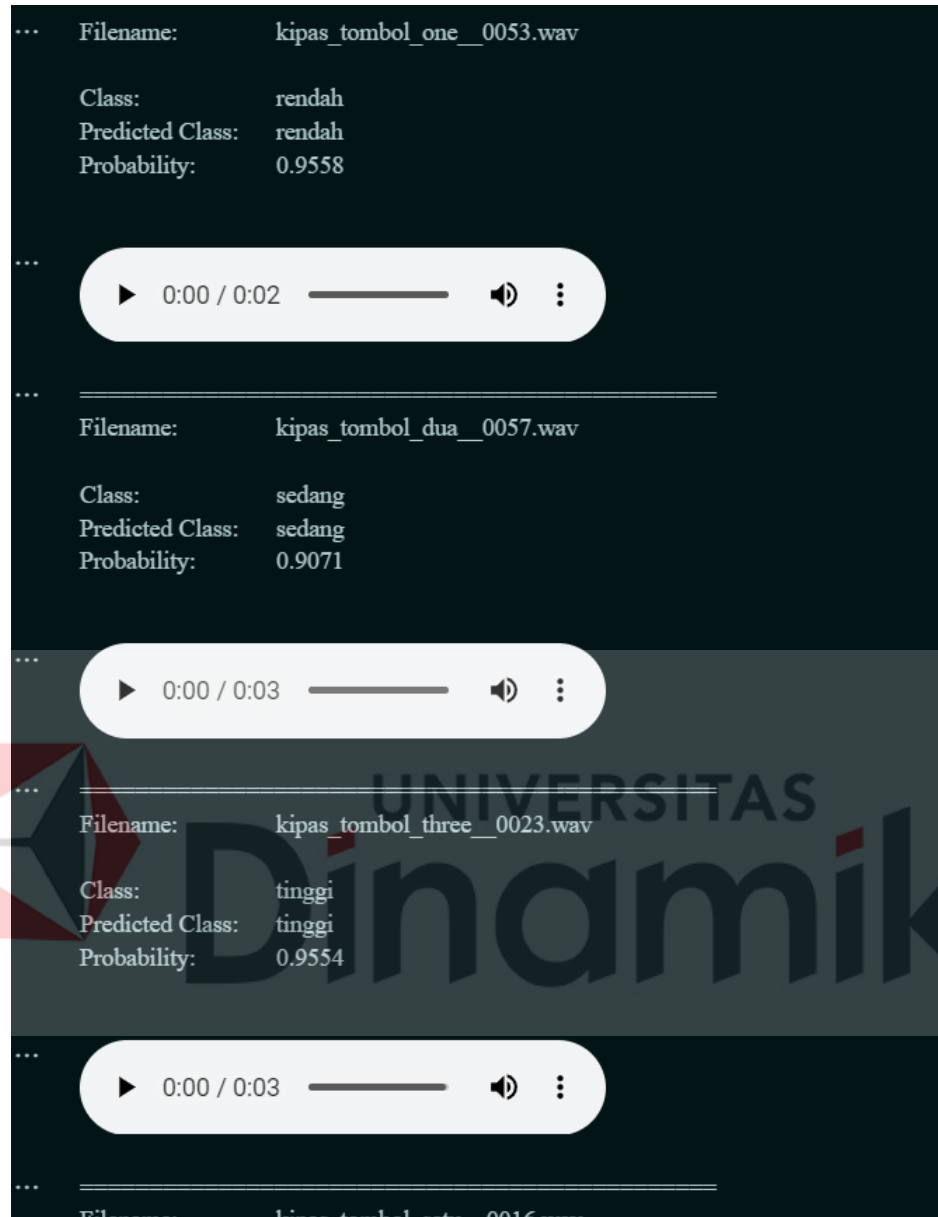
terhadap aspek tertentu dalam dataset. Hal ini memerlukan pemahaman lebih lanjut terkait faktor – faktor yang mempengaruhi nilai Matthews, sehingga dapat diidentifikasi dan diatasi secara efektif. Analisis mendalam terhadap perbedaan ini menjadi kunci untuk mengoptimalkan kinerja model, dengan fokus pada memahami Dinamika yang mendasari perbedaan nilai Matthews dibandingkan metrik lainnya. Dengan demikian, hasil ini memberikan wawasan penting yang dapat membimbing langkah – langkah perbaikan dan fine-tuning model untuk mencapai konsistensi dan kehandalan yang lebih tinggi dalam menghadapi variasi situasi. Oleh karena itu, pemahaman lebih mendalam terhadap implikasi metrik Matthews menjadi langkah penting dalam merinci dan memahami aspek – aspek tertentu yang perlu diperhatikan dan diperbaiki guna meningkatkan keseluruhan performa model.



Gambar 4.13 *Model Performance Metric Model Kedua*

Pada pengujian model kedua, penerapan pada dataset subset test menunjukkan kinerja yang sangat memuaskan, sebagaimana tercermin pada Gambar 4.14. hasil klasifikasi yang diperoleh sesuai dengan prediksi model, menciptakan gambaran yang positif terkait

kehandalan model dalam menghadapi data yang belum pernah dilihat sebelumnya. Pengamatan ini memberikan indikasi bahwa model kedua mampu mengatasi tantangan dan variabilitas yang mungkin muncul dalam situasi dunia nyata. Hasil yang sangat baik pada pengujian ini dapat diartikan sebagai kemajuan yang signifikan dari model sebelumnya. Kesesuaian antara hasil klasifikasi dan prediksi menunjukkan kemampuan model dalam menangkap pola yang kompleks dan variative pada subset test. Hal ini memvalidasi efektivitas model dalam menggeneralisasi pembelajaran dari data pelatihan ke data uji yang belum pernah dilihat sebelumnya. Keberhasilan ini juga memberikan gambaran bahwa model dapat diandalkan dalam berbagai konteks praktis. Hasil yang mencerminkan kualitas prediksi yang sangat baik pada gambaran keseluruhan memberikan keyakinan bahwa model kedua dapat dijadikan sebagai solusi yang efektif dalam menerapkan pemahaman dan pembelajaran yang diperolehnya. Perbandingan dengan hasil model sebelumnya menunjukkan peningkatan yang substansial, memperkuat argument bahwa perbaikan dan iterasi pada model mampu menghasilkan peningkatan kinerja yang signifikan. Dengan demikian, eksperimen ini mengonfirmasi bahwa model kedua lebih handal dan dapat diandalkan untuk diterapkan dalam konteks dunia nyata, seiring dengan kemampuannya memberikan hasil klasifikasi yang sangat baik pada dataset *subset test*.



Gambar 4.14 Pengujian Model Kedua Menggunakan Dataset *Test*

Dalam proses pengujian klasifikasi audio menggunakan model kedua yang dilakukan sebanyak 40 kali, dimana pengujian tersebut mencakup seluruh perintah yang ada dan pengujian setiap masing – masing perintah dilakukan sebanyak 10 kali untuk mengatur kipas angin yang diuji untuk memastikan respon model yang akurat dan konsisiten terhadap setiap perintah level dalam kecepatan kipas angin. Pengujian ini bertujuan untuk

mengevaluasi kemampuan model dalam memahami konteks spesifik dari setiap perintah dan respon yang sesuai.

Tabel 4.4 Pengujian Dataset Subset Test Menggunakan Model Dataset Kedua

BAHASA	STATUS KIPAS	AKURASI	
		BENAR	SALAH
Bahasa Indonesia	0	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	1	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	2	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	3	4	0
Bahasa Jawa		3	0
Bahasa Inggris		2	1
Jumlah		39	1
Akurasi		97,5%	2,5%

Hasil pengujian menggunakan model yang telah mengalami pelatihan dengan 15 *epoch*, diketahui bahwa tingkat akurasi mencapai 97,5%. Hasil tersebut mencerminkan kinerja yang sangat baik dalam kemampuan klasifikasi audio. Meskipun demikian, perlu dicatat bahwa terdapat beberapa kasus dimana model tidak sepenuhnya sesuai dalam melakukan klasifikasi audio. Meskipun angka akurasi yang tinggi menunjukkan kehandalan model secara umum penting untuk menyoroiti beberapa percobaan dimana

model mungkin mengalami tantangan atau kegagalan dalam mengklasifikasikan data audio dengan akurat.

Tabel 4.5 Pengujian Pengguna Suara Yang Terdapat Dalam Model Dataset Kedua

NAMA PENGGUNA	BAHASA	STATUS KIPAS	AKURASI		KETERANGAN
			BENAR	SALAH	
Pegguna Yang Terdapat Dalam Dataset	Kipas Mati	0	7	3	Akurasi benar = 60% Akurasi salah = 40%
	Kipas Pejah		5	5	
	Kipas Off		6	4	
	Jumlah		18	12	
	Kipas Tombol Satu	1	6	4	Akurasi benar = 60% Akurasi salah = 40%
	Kipas Tombol Siji		7	3	
	Kipas Tombol One		5	5	
	Jumlah		18	12	
	Kipas Tombol Dua	2	7	3	Akurasi benar = 60,7% Akurasi salah = 33,3%
	Kipas Tombol Loro		7	3	
	Kipas Tombol Two		6	4	
	JUMLAH		20	10	
	Kipas Tombol Tiga	3	7	3	Akurasi benar = 56,7% Akurasi salah = 43,3%
	Kipas Tombol Telu		4	6	
	Kipas Tombol Three		5	5	
JUMLAH		17	13		

Hasil uji klasifikasi suara pengguna yang terdapat dalam dataset merupakan suara pengguna tersebut juga sudah terdapat dalam dataset yang telah melewati proses training dengan 15 epoch *epoch* menunjukkan bahwa tingkat akurasi yang dicapai bervariasi untuk setiap status. Status 0 dan 1 memiliki tingkat akurasi benar sebesar 60%, dengan tingkat akurasi salah sebesar 40%. Sementara itu, status 2 menunjukkan peningkatan sedikit dalam akurasi benar, mencapai 60,7%, dengan tingkat akurasi salah sebesar 33,3%. Di sisi lain,

status 3 menunjukkan tingkat akurasi benar sebesar 56,7%, dengan tingkat akurasi salah sebesar 43,3%.

Tabel 4.6 Pengujian Suara Pengguna Yang Tidak Terdapat Dalam Model Dataset Kedua

NAMA PENGGUNA	BAHASA	STATUS KIPAS	AKURASI		KETERANGAN
			BENAR	SALAH	
Pegguna Suara Yang Tidak Terdapat Dalam Dataset	Kipas Mati	0	4	6	Akurasi benar = 40% Akurasi salah = 60%
	Kipas Pejah		3	7	
	Kipas Off		5	5	
	Jumlah		12	18	
	Kipas Tombol Satu	1	4	6	Akurasi benar = 43,3% Akurasi salah = 56,7%
	Kipas Tombol Siji		4	6	
	Kipas Tombol One		5	5	
	Jumlah		13	17	
	Kipas Tombol Dua	2	6	4	Akurasi benar = 50% Akurasi salah = 50%
	Kipas Tombol Loro		5	5	
	Kipas Tombol Two		4	6	
	Jumlah		15	15	
	Kipas Tombol Tiga	3	7	3	Akurasi benar = 40% Akurasi salah = 60%
	Kipas Tombol Telu		2	8	
Kipas Tombol Three	3		7		
Jumlah		12	18		

Berdasarkan hasil evaluasi klasifikasi suara pada pengguna yang tidak ada dalam dataset adalah suara pengguna yang tidak tersedia di dalam dataset pada saat proses pelatihan data dengan 15 epoch yang dapat diamati bahwa kinerja model bervariasi tergantung pada status pengguna. Untuk pengguna dengan status 0, model menunjukkan tingkat akurasi benar sebesar 40%, sementara tingkat akurasi salah mencapai 60%. Pada status 1, meskipun terdapat peningkatan sedikit, tingkat akurasi benar sebesar 43,3%, namun tingkat akurasi salah masih tinggi, yakni 56,7%. Pengguna dengan status 2

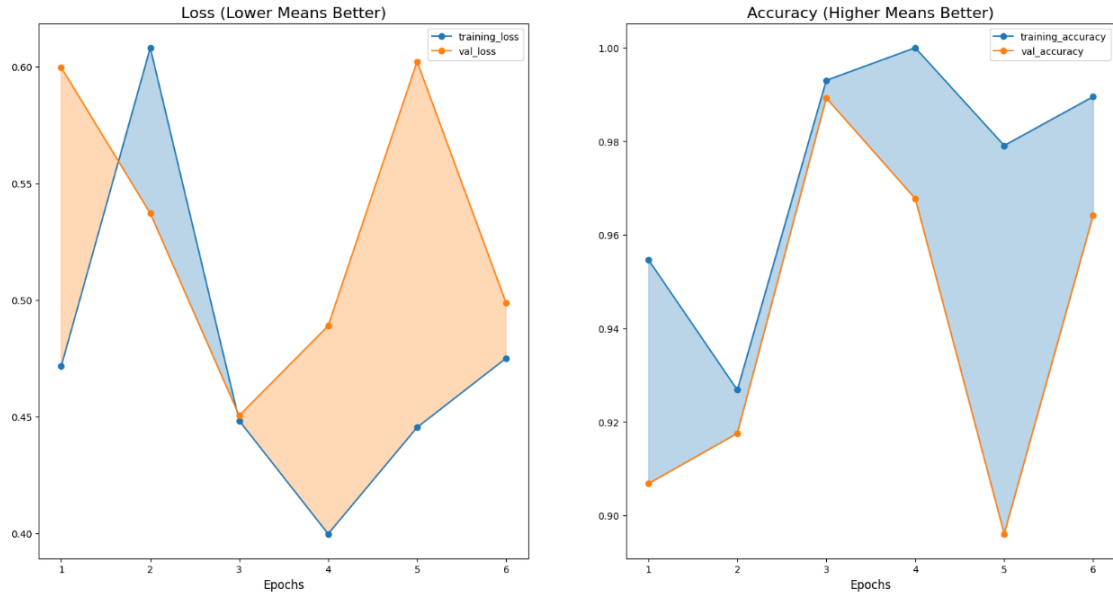
menunjukkan hasil yang lebih seimbang dengan tingkat akurasi benar sebesar 50% dan tingkat akurasi salah sebesar 50%. Sementara itu, pada status 3, model kembali menunjukkan tingkat akurasi benar sebesar 40%, dengan tingkat akurasi salah yang mencapai 60%.

### 4.3 Hasil Proses Training Model Ketiga Dengan 15 Epoch Early Stopping

Proses pelatihan model ketiga berhasil mencapai akurasi sempurna sebesar 1.00 dan *loss* sebesar 0.46, sejalan dengan hasil pelatihan sebelumnya. Namun, terdapat perbedaan signifikan pada pelatihan ketiga ini, yaitu pada *epoch* ke – 5, proses training dihentikan secara otomatis. Hal ini disebabkan adanya fitur *early stopping* dalam program, yang berfungsi untuk menghentikan proses training jika akurasi tidak mengalami perubahan yang signifikan. Gambar 4.15 menggambarkan grafik akurasi selama proses pelatihan, menunjukkan adanya peningkatan dan penurunan yang mencirikan aktivasi fitur *early stopping*.

Ketika model mencapai akurasi maksimal pada 1.00, algoritma *early stopping* secara otomatis mengidentifikasi bahwa tidak ada peningkatan signifikan dalam kinerja model setelah *epoch* ke – 5. Dengan demikian, proses training dihentikan untuk menghindari *overfitting* dan memastikan bahwa model tidak hanya menghafal data pelatihan tanpa kemampuan generalisasi yang baik pada data baru. Meskipun akurasi maksimal telah tercapai, penggunaan *early stopping* memberikan keuntungan tambahan dalam mencegah potensi *overtraining*.

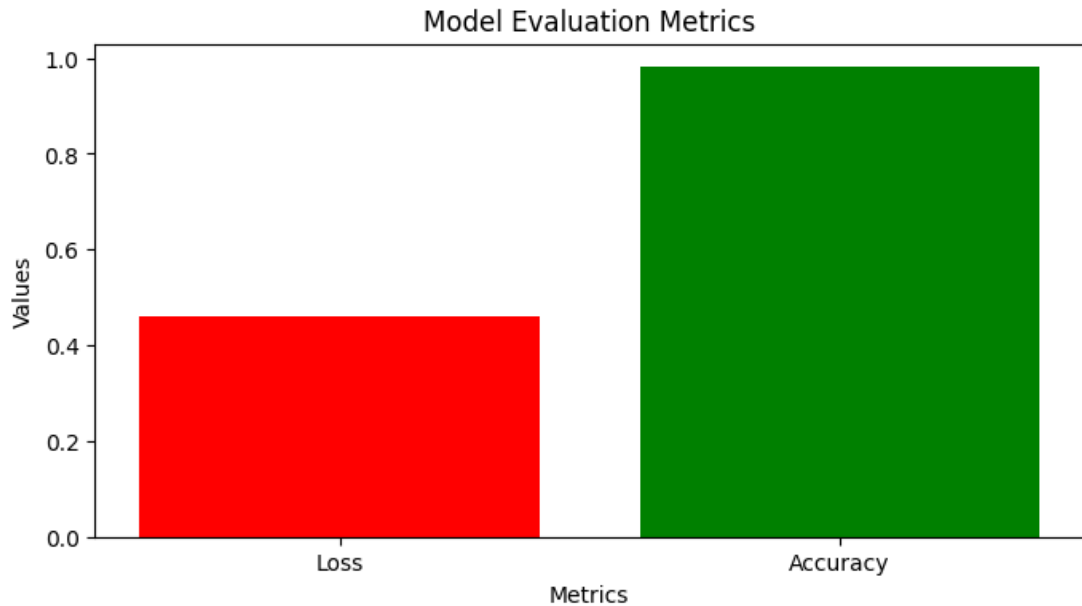
Hasil ini menyoroti keefektifan fitur *early stoping* dalam memantau dalam mengoptimalkan proses pelatihan model. sebagai hasil dari strategi ini, diperoleh model dengan akurasi tinggi yang dihasilkan dengan efisiensi waktu, mengurangi risiko *overfitting*, dan memastikan bahwa model dapat diterapkan secara optimal pada data baru. Fitur *early stopping* menjadu elemen kunci dalam menjaga keseimbangan antara akurasi dan generalisasi model, mendukung keandalan dan keterampilan model dalam menghadapi tantangan dunia nyata.



Gambar 4.15 Akurasi dan *Loss* Model Ketiga

Pada evaluasi model menghasilkan akurasi dan *loss* sama seperti pada model kedua hanya saja pada model ketiga ini pada *loss*nya menghasilkan 0.51. hasil ini menandakan kemajuan yang positif dalam menangani kompleksitas dataset. Dengan akurasi yang serupa, penurunan nilai *loss* pada model ketiga memberikan indikasi bahwa model tersebut mampu mengoptimalkan prediksi dengan tingkat ketidakpastian yang lebih rendah. Nilai *loss* yang lebih kecil menggambarkan bahwa model mampu mengenali pola dan hubungan antar fitur dengan lebih efisien, menciptakan representasi internal yang lebih akurat. Percobaan ini dapat disebabkan oleh penyesuaian pada arsitektur mode, *hyperparameter*, atau *training* yang perbaikan pada nilai *loss* tidak mengorbankan kinerja akurasi. Hasil ini juga menyoroti keberhasilan upaya peningkatan model seiring dengan perkembangan *iterative*. Selain itu, perbandingan kinerja antara model – model tersebut dapat memberikan wawasan tambahan terkait *trade – off* antara akurasi dan *loss* dalam konteks klasifikasi. Oleh karena itu, penelitian lebih lanjut diperlukan untuk memahami secara mendalam faktor – faktor yang menyebabkan perbedaan kinerja antara model kedua dan ketiga. Kesimpulannya, peningkatan pada nilai *loss* kompleksitas data, walaupun diperlukan analisis lebih lanjut untuk memahami dampaknya terhadap kinerja keseluruhan dan menentukan apakah perubahan ini membawa manfaat signifikan pada konteks aplikatif.





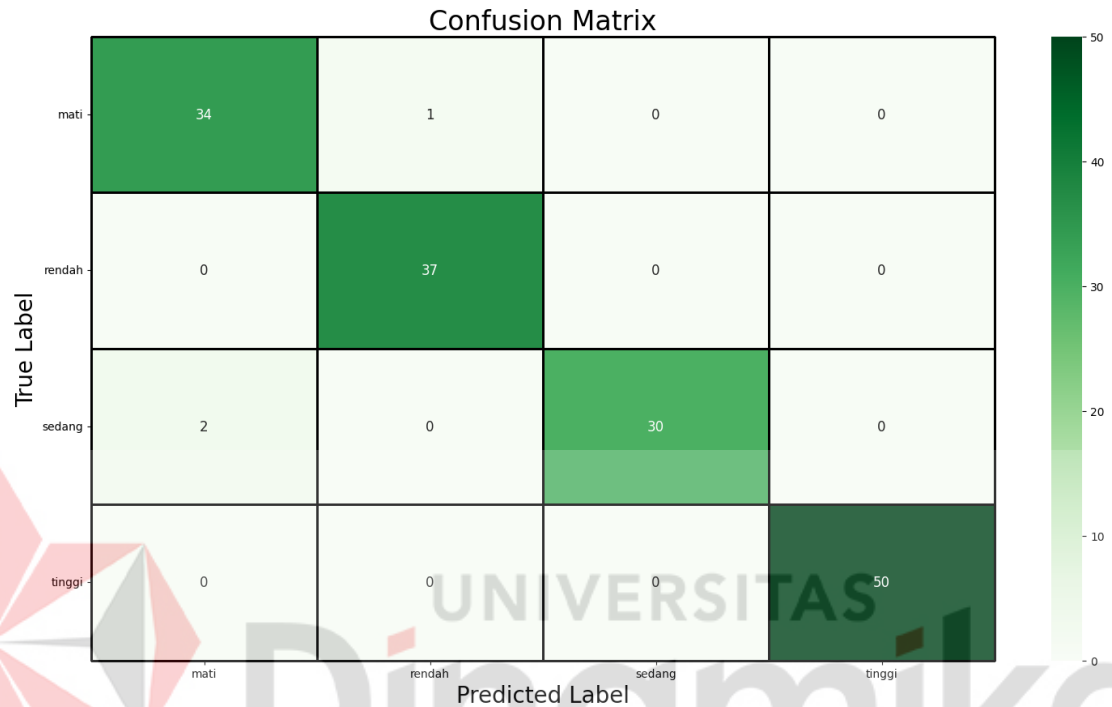
Gambar 4.16 Evaluasi Model

Hasil prediksi model ketiga, seperti yang terungkap melalui *confusion matrix*, menunjukkan sebagian besar hasil prediksi yang konsisten antara kelas dan prediksi, mencerminkan tingkat akurasi yang baik. Namun, ditemukan beberapa ketidaksesuaian pada beberapa kelas tertentu. Sebagai contoh, pada kelas “mati”, terdapat satu kasus yang diprediksi sebagai kelas “rendah”, sementara pada kelas “sedang”, terdapat dua kasus yang seharusnya menjadi kelas “mati” berdasarkan hasil prediksinya.

Ketidaksesuaian ini dapat memberikan wawasan berharga terkait dengan kelemahan spesifik model, serta potensi tantangan dalam mengkalasifikasikan kelas tertentu. Oleh karena itu, diperlukan evaluasi lebih lanjut terhadap faktor – faktor yang mungkin mempengaruhi ketidaksesuaian ini, seperti karakteristik data yang kompleks atau kekurangan representasi kelas tertentu dalam dataset.

Meskipun demikian, penting untuk diingat bahwa hasil ini tidak hanya mencerminkan kekurangan model, tetapi juga dapat memberikan pandangan yang berguna untuk perbaikan lebih lanjut. Dengan mengetahui ketidaksesuaian kelas tertentu, langkah – langkah penyesuaian dan tuning dapat diarahkan untuk meningkatkan kinerja model, khususnya dalam mengatasi kesulitan mengklasifikasi kelas “mati” dan “sedang”. Hasil ini,

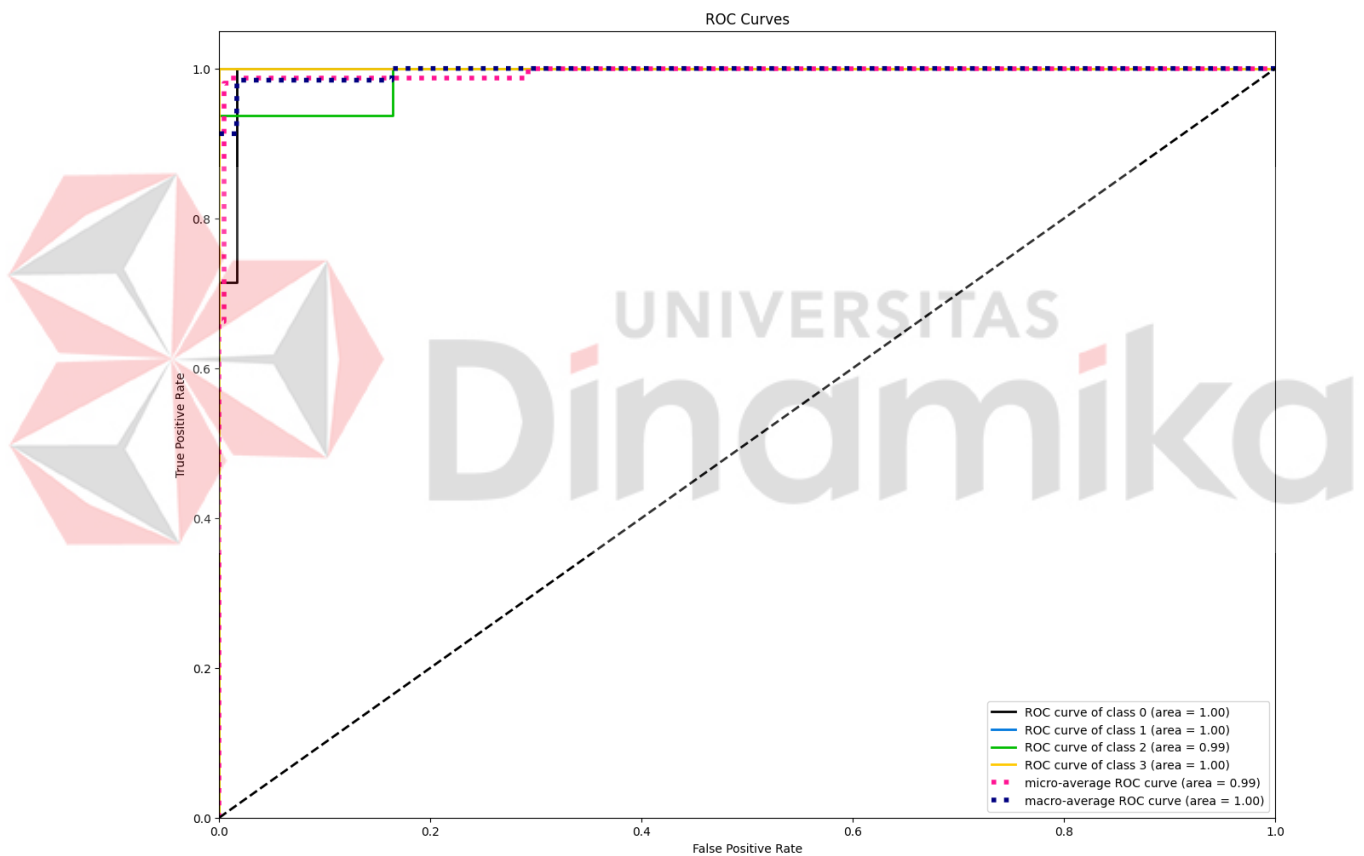
bagaimanapun, memberikan dasar yang solid untuk penelitian lebih lanjut dan pengembangan model, dengan tujuan untuk meningkatkan akurasi dan keandalan dalam menghadapi situasi dunia nyata.



Gambar 4.17 *Confusion Matrix* Model Ketiga

Hasil analisis pada *ROC Curve* model pelatihan ketiga menggambarkan bahwa terdapat satu kelas yang mencapai nilai 0.99, memperlihatkan perbedaan sebesar 0.1 dari tiga kelas lainnya yang mencapai akurasi puncak, yaitu 1.00. temuan ini mengindikasikan ketepatan model dalam memisahkan kelas – kelas tersebut. *ROC Curve* memeberikan gambaran *visual* yang jelas tentang trade – off antara *true positive rate* dan *false positive rate*, dan tingginya nilai yang dicapai oleh empat dari empat kelas memberikan indikasi kuat terkait kemampuan model dalam membedakan dengan sangat baik antara kelas positif dan negatif. Adanya satu kelas yang memiliki nilai 0.99 menunjukkan bahwa model cenderung memberikan prediksi yang sangat akurat untuk kelas tersebut, walaupun perlu diperhatikan bahwa terdapat sedikit selisih dengan tiga kelas lainnya yang mencapai tingkat akurasi maksimal. Hal ini dapat menjadi titik fokus untuk pemeriksaan lebih lanjut guna

memastikan bahwa perbedaan tersebut bukanlah hasil dari variabilitas acak atau fenomena yang tidak diinginkan. Selain itu, tingginya akurasi pada tiga kelas lainnya menunjukkan bahwa model secara umum memiliki kemampuan yang sangat baik dalam mengklasifikasikan data dengan benar. Oleh karena itu, walaupun terdapat sedikit selisih pada satu kelas, hasil ini menegaskan kemampuan model ketiga dalam melakukan klasifikasi yang sangat baik secara keseluruhan, dan dapat menjadi dasar untuk pertimbangan lebih lanjut terkait peningkatan atau fine – tuning pada aspek – aspek spesifik yang mungkin mempengaruhi performa model pada class tertentu.

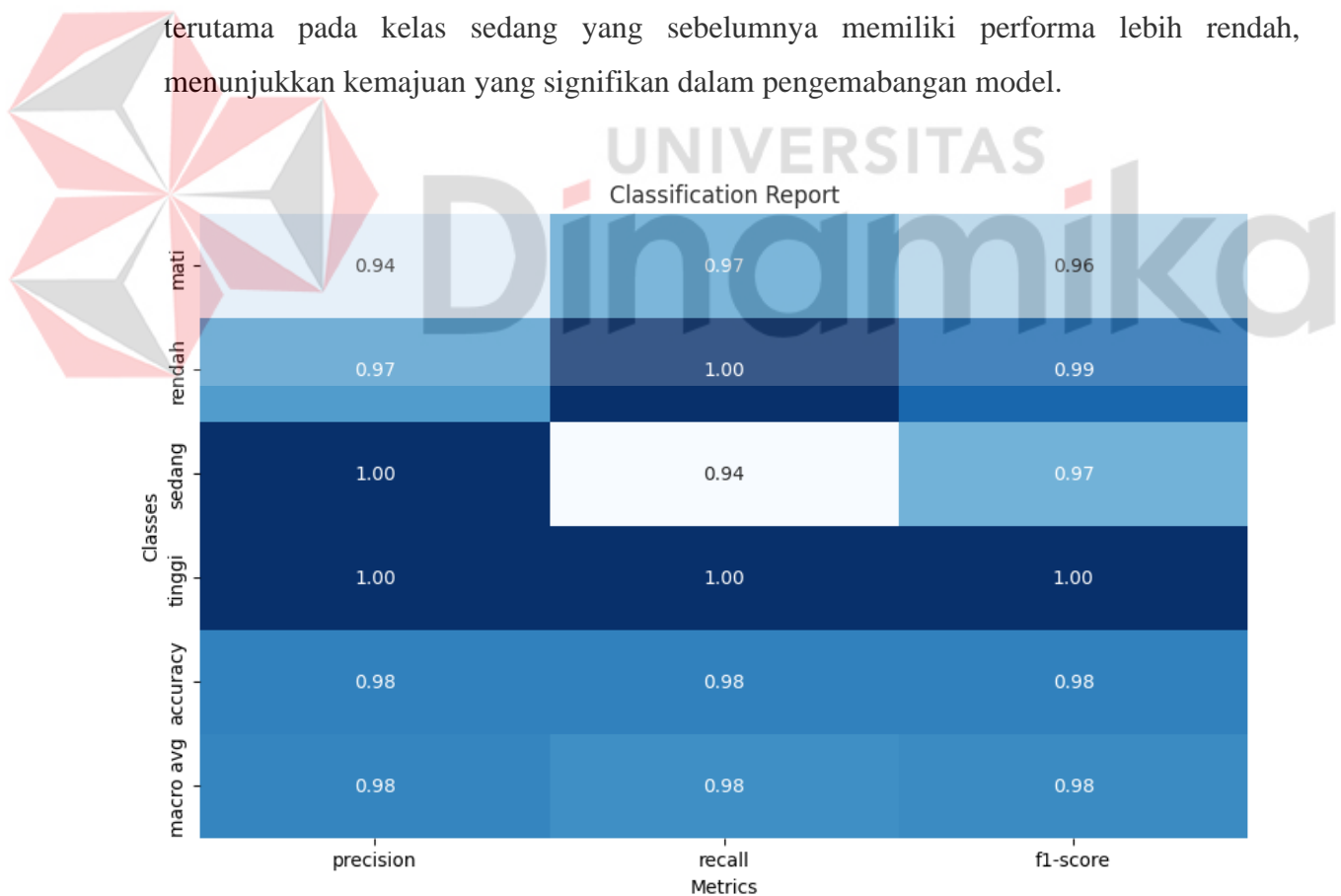


Gambar 4.18 ROC Curve Model Data Ketiga

Dalam hasil *classification report*, terlihat bahwa perhitungan menggunakan precision pada kelas mati menunjukkan akurasi sebesar 94%, sementara kelas rendah dan tinggi memiliki akurasi 97% dan kelas sedang mencapai akurasi 100%. Pada perhitungan recall, kelas tinggi dan rendah mendapatkan akurasi 100%, sementara kelas mati dan sedang

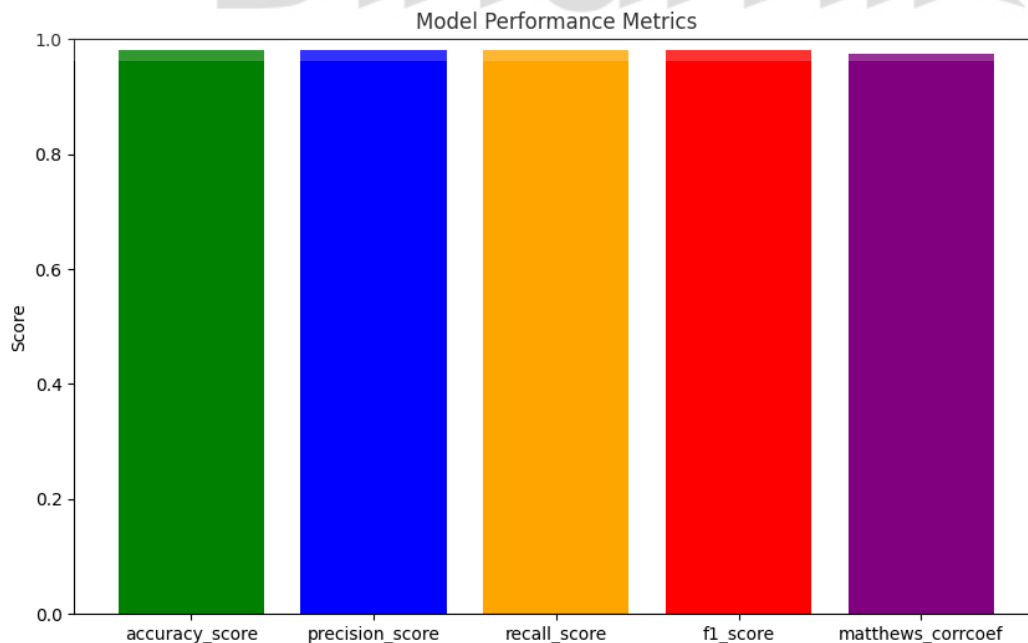
memiliki nilai akurasi masing – masing sebesar 97% dan 94%. Pentingnya evaluasi model juga tercermin pada perhitungan *f1 – score*, dimana hanya kelas tinggi yang mencapai akurasi 100%, sementara tiga *class* lainnya memiliki nilai diatas 95%. secara keseluruhan, ketiga metrik evaluasi tersebut menghasilkan akurasi sebesar 98% untuk setiap kelas dan rata – rata makro mencapai nilai 98%.

Hasil ini memberikan gambaran positif tentang kemampuan model dalam melakukan prediksi dengan tingkat akurasi yang tinggi pada setiap kelas. Khususnya, kelas tinggi mencapai akurasi maksimal dalam *precision*, *recall* dan *f1 – score*, menunjukkan kemampuan model dalam mengenali dan mengklasifikasikan instance pada kelas tersebut dengan sangat baik. Meskipun demikian, perlu diingat bahwa evaluasi model tidak hanya bergantung pada nilai akurasi, namun juga mempertimbangkan karakteristik masing – masing *class* dalam konteks tugas klasifikasi. Peningkatan akurasi pada semua *class*, terutama pada kelas sedang yang sebelumnya memiliki performa lebih rendah, menunjukkan kemajuan yang signifikan dalam pengembangan model.



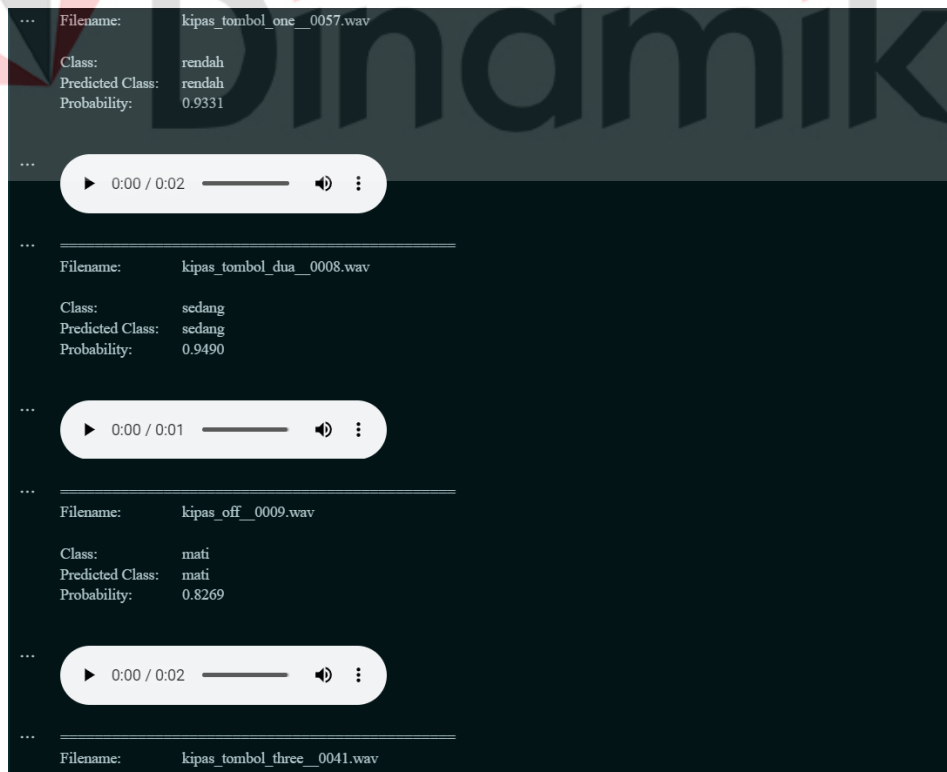
Gambar 4.19 *Classification Report* Model Data Ketiga

Selanjutnya, dalam analisis model pada Gambar 4.20, metrik kinerja menunjukkan bahwa seluruh perhitungan menghasilkan nilai yang identik. Hal ini mencerminkan hasil konsisten dari berbagai aspek evaluasi kinerja model. Meskipun semua metrik menunjukkan nilai yang sama, penting untuk menggali lebih dalam guna memahami penyebab keseragaman ini. Analisis lebih lanjut mungkin diperlukan untuk mengeksplorasi faktor – faktor yang berkontribusi terhadap konsistensi tersebut, baik dari segi parameter model maupun karakteristik dataset yang digunakan. Kondisi ini dapat memicu pertanyaan tentang apakah model tersebut mencapai titik konvergensi atau ada elemen – elemen lain yang memengaruhi hasil secara seragam. Meskipun hasil yang konsisten dapat dianggap positif, interpretasi lebih lanjut diperlukan untuk memastikan bahwa tidak ada faktor eksternal atau kebijakan model yang menyebabkan hasil seragam ini. Oleh karena itu, sementara keseragaman dapat dianggap sebagai keberhasilan, perlu juga pemahaman mendalam terhadap faktor yang mungkin memengaruhi hal tersebut. Analisis lebih lanjut terhadap model performance metric pada Gambar 4.20 dapat memberikan wawasan yang lebih lengkap dan kontekstual terhadap hasil yang mencerminkan keseragaman di seluruh evaluasi kinerja model.



Gambar 4.20 Model Performance Metric Model Data Ketiga

Pada tahap akhir penelitian, dilakukan pengujian model menggunakan dataset dari *subset test*, dan hasilnya tercermin dalam Gambar 4.21. Hasil pengujian tersebut menunjukkan bahwa pengklasifikasian yang dilakukan oleh model sesuai dengan prediksi yang telah dihasilkan. Gambar 4.21 memberikan gambaran visual tentang kecocokan antara kelas actual dan prediksi model pada dataset uji. Hasil yang sesuai ini memberikan indikasi positif terhadap kemampuan model untuk menggeneralisasi dan melakukan prediksi dengan akurat pada data yang belum pernah dilihat sebelumnya. Keberhasilan pengklasifikasian yang tergambar pada Gambar 4.21 memberikan keyakinan tambahan terhadap keandalan model dalam mengatasi variasi dan kompleksitas data di dunia nyata. Penting untuk dicatat bahwa subset test memberikan tantangan yang berbeda dibandingkan dengan dataset pelatiba, sehingga keberhasilan model pada tahap ini menggambarkan kemampuannya dalam menghadapi situasi uji yang mewakili kondisi dunia nyata.



Gambar 4.21 Pengujian Model Data Ketiga Menggunakan *Subset Test*

Dalam proses pengujian klasifikasi audio menggunakan model ketiga yang dilakukan sebanyak 40 kali, dimana pengujian tersebut mencakup seluruh perintah yang ada dan pengujian setiap masing – masing perintah dilakukan sebanyak 10 kali untuk mengatur kipas angin yang diuji untuk memastikan respon model yang akurat dan konsisiten terhadap setiap perintah level dalam kecepatan kipas angin. Pengujian ini bertujuan untuk mengevaluasi kemampuan model dalam memahami konteks spesifik dari setiap perintah dan respon yang sesuai.

Tabel 4.7 Pengujian Dataset Subset Test Menggunakan Model Ketiga

BAHASA	STATUS KIPAS	AKURASI	
		BENAR	SALAH
Bahasa Indonesia	0	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	1	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	2	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Bahasa Indonesia	3	4	0
Bahasa Jawa		3	0
Bahasa Inggris		3	0
Jumlah		40	0
Akurasi		100%	0%

hasil pengujian pada subset test dataset menggunakan model yang telah dilatih menggunakan 15 *epoch* dengan *early-stopping* menunjukkan tingkat akurasi sebesar 100%. Capaian ini menandakan bahwa model tersebut telah melakukan klasifikasi dengan sangat baik, mencapai tingkat akurasi yang optimal dalam membedakan data audio. Dengan akurasi penuh ini, model telah berhasil mengklasifikasikan setiap sampel data uji dengan benar, memberikan keyakinan bahwa model memiliki kemampuan yang sangat baik dalam memahami dan membedakan ciri – ciri audio yang kompleks. Hasil tersebut memberikan keyakinan bahwa model tersebut mampu mengidentifikasi pola – pola audio dengan presisi tinggi, bahkan dalam situasi yang kompleks sekalipun. Kinerja yang sangat baik ini menunjukkan bahwa pelatihan model melalui 15 epoch dengan *early – stopping* telah menghasilkan representasi internal yang optimal dari data audio, memungkinkan model untuk membuat Keputusan klasifikasi yang tepat secara konsisten.

Tabel 4.8 Pengujian Suara Pengguna Yang Terdapat Dalam Dataset Melalui Mikروفon Menggunakan Model Ketiga

NAMA PENGGUNA	BAHASA	STATUS KIPAS	AKURASI		KETERANGAN	
			BENAR	SALAH		
Pengguna Suara Yang Terdapat Dalam Dataset	Kipas Mati	0	6	4	Akurasi benar = 56,7% Akurasi salah = 43,3%	
	Kipas Pejah		5	5		
	Kipas Off		6	4		
	Jumlah			17	13	
	Kipas Tombol Satu	1	7	3	Akurasi benar = 66,7% Akurasi salah = 33,3%	
	Kipas Tombol Siji		7	3		
	Kipas Tombol One		6	4		
	Jumlah			20	10	
	Kipas Tombol Dua	2	6	4	Akurasi benar = 56,7% Akurasi salah = 43,3%	
	Kipas Tombol Loro		5	5		
	Kipas Tombol Two		6	4		
	Jumlah			17	13	



	Kipas Tombol Tiga	3	6	4	Akurasi benar = 50% Akurasi salah = 50%
	Kipas Tombol Telu		4	6	
	Kipas Tombol Three		5	5	
	Jumlah	15	15		

Berdasarkan hasil pengujian klasifikasi suara pada pengguna dalam dataset merupakan suara pengguna yang tersedia dalam dataset atau yang sudah dilatih pada proses pelatihan dengan 15 *epoch early – stopping* ketika tidak ada perubahan akurasi yang banyak maka dari fitur *early - stopping* akan memberhentikan proses pelatihan meskipun tidak sampai pada epoch 15. Hasil dari pengujian ini dapat ditemukan bahwa klasifikasi berdasarkan status menunjukkan variasi dalam akurasi prediksi. Status 0 memiliki akurasi benar sebesar 56,7% dan akurasi salah sebesar 43,3%, sementara status 1 memiliki tingkat akurasi benar sebesar 66,7% dengan tingkat akurasi salah 33,3%. Pengguna dengan status 2 menunjukkan akurasi benar sebesar 56,7% dan akurasi salah sebesar 43,3%, sedangkan status 3 menunjukkan tingkat akurasi benar sebesar 50%, dengan tingkat akurasi salah yang sama. Pengujian ini mengindikasikan bahwa model klasifikasi suara cenderung memberikan prediksi yang lebih akurat untuk pengguna dengan status 1, sementara pengguna dengan status 3 menunjukkan tingkat akurasi yang sebanding antara prediksi benar dan salah.

Tabel 4.9 Pengujian Klasifikasi Suara Pengguna Yang Tidak Terdapat Dalam Dataset Melalui Mikrofon Menggunakan Model Ketiga

NAMA PENGGUNA	BAHASA	STATUS KIPAS	AKURASI		KETERANGAN
			BENAR	SALAH	
Pengguna Suara Yang Tidak Terdapat Dalam Dataset	Kipas Mati	0	5	5	Akurasi benar = 53,3% Akurasi salah = 46,7%
	Kipas Pejah		4	6	
	Kipas Off		7	3	
	Jumlah		16	14	
	Kipas Tombol Satu	1	5	5	Akurasi benar = 43,3% Akurasi salah = 56,7%
	Kipas Tombol Siji		4	6	
	Kipas Tombol One		4	6	

	Jumlah		13	17	
	Kipas Tombol Dua	2	4	6	Akurasi benar = 36,7% Akurasi salah = 63,3%
	Kipas Tombol Loro		4	6	
	Kipas Tombol Two		3	7	
	JUMLAH		11	19	
	Kipas Tombol Tiga	3	7	3	Akurasi benar = 43,7% Akurasi salah = 56,7%
	Kipas Tombol Telu		2	8	
	Kipas Tombol Three		4	6	
	Jumlah		13	17	

Berdasarkan hasil pengujian klasifikasi suara pada pengguna tidak terdapat dalam dataset merupakan suara pengguna yang tidak ada dalam dataset atau yang tidak termasuk kedalam proses pelatihan dengan *15 epoch early – stopping* dimana *early – stopping* ketika tidak ada perubahan akurasi tidak terlalu jauh berbeda dari akurasi sebelumnya maka dari fitur *early – stopping* akan memberhentikan proses pelatihan meskipun tidak sampai pada *epoch 15*, ditemukan bahwa status 0 memiliki tingkat akurasi benar sebesar 53,3%, yang berarti model berhasil mengidentifikasi dengan benar sebanyak itu dari keseluruhan sampel pada kategori ini. Di sisi lain, tingkat akurasi salahnya mencapai 46,7%, menunjukkan bahwa sekitar 46,7% dari sampel dalam kategori ini salah diklasifikasikan oleh model. Status 1 menunjukkan tingkat akurasi benar sebesar 43,3%, yang menandakan bahwa model berhasil mengklasifikasikan dengan benar sekitar 43,3% dari sampel dalam kategori tersebut. Meskipun demikian, tingkat akurasi salahnya mencapai 56,7%, menandakan bahwa sekitar 56,7% dari sampel dalam kategori ini diklasifikasikan secara tidak tepat. Situasi serupa terjadi pada status 2, di mana tingkat akurasi benar mencapai 36,7%, sementara tingkat akurasi salah mencapai 63,3%. Status 3 juga menunjukkan hasil yang sebanding, dengan tingkat akurasi benar sebesar 43,7% dan tingkat akurasi salah mencapai 56,7%.

#### 4.4 Perbandingan Hasil Pengujian Antar Epoch

Dari hasil pengujian, akurasi klasifikasi suara kipas angin meningkat seiring dengan bertambahnya *epoch*. Namun, model YAMNet pada *epoch 15 early stopping* cenderung mengklasifikasikan suara kipas angin yang mati sebagai suara kipas angin yang hidup. Kemungkinan penyebabnya adalah model YAMNet terlalu dilatih pada data pelatihan.

Secara umum, *epoch 15* memberikan akurasi yang lebih baik untuk semua perintah kipas, kecuali perintah “mati” dan “off”. Untuk perintah “mati” dan “off” pada *epoch 12* memberikan akurasi yang lebih baik.

Tabel 4.10 Perbandingan Akurasi Pengguna Yang Terdapat Dalam Dataset Antar Epoch

perintah	epoch 12		epoch 15		epoch 15 (early stopping)	
	akurasi benar	akurasi salah	akurasi benar	akurasi salah	akurasi benar	akurasi salah
kipas mati	43,3%	56,7%	60%	40%	56,7%	43,3%
kipas off						
kipas pejah						
kipas tombol satu	40%	60%	60%	40%	66,7%	33,3%
kipas tombol one						
kipas tombol siji						
kipas tombol dua	60%	40%	60,7%	33,3%	56,7%	43,3%
kipas tombol two						
kipas tombol loro						

kipas tombol tiga						
kipas tombol three	43,3%	56,7%	56,7%	43,3%	50%	50%
kipas tombol telu						

Berdasarkan Tabel 4.10 bahwa akurasi benar untuk semua perintah kipas meningkat seiring dengan bertambahnya *epoch*. Namun, akurasi benar untuk perintah “mati” dan “off” pada *epoch 15 early stopping* justru menurun dibandingkan dengan *epoch 15*. Hal ini menunjukkan bahwa model YAMNet pada *epoch 15 early stopping* lebih sering mengklasifikasikan suara kipas angin yang mati sebagai suara kipas angin yang hidup. Kemungkinan penyebabnya adalah model YAMNet telah terlalu dilatih pada data pelatihan, sehingga menjadi terlalu sensitif terhadap pola – pola yang ada pada data pelatihan. Akibatnya, model YAMNet menjadi lebih sulit untuk mengklasifikasikan suara kipas angin yang berbeda dari data pelatihan.

Tabel 4.11 Perbandingan Akurasi Pengguna Yang Tidak Terdapat Dalam Dataset Antar *Epoch*

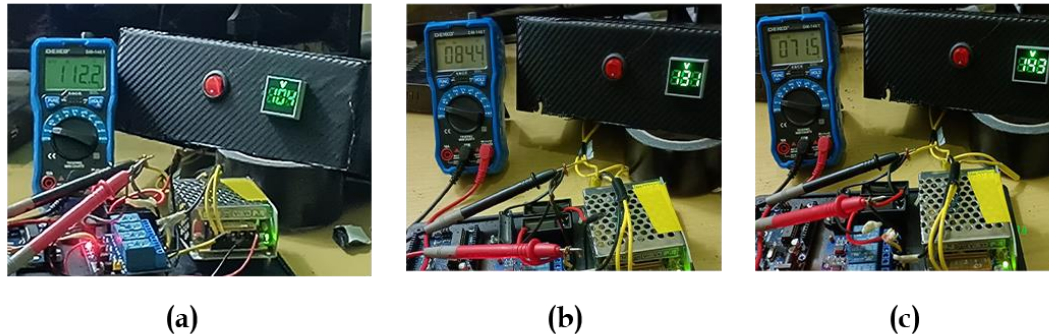
perintah	<i>epoch 12</i>		<i>epoch 15</i>		<i>epoch 15 (early stopping)</i>	
	akurasi benar	akurasi salah	akurasi benar	akurasi salah	akurasi benar	akurasi salah
kipas mati	10%	90%	40%	60%	53,3%	46,7%
kipas off						
kipas pejah						

kipas tombol satu						
kipas tombol one	30%	70%	43,3%	56,7%	43,3%	56,7%
kipas tombol siji						
kipas tombol dua						
kipas tombol two	56,7%	43,3%	50%	50%	36,7%	63,3%
kipas tombol loro						
kipas tombol tiga						
kipas tombol three	20%	80%	40%	60%	43,7%	56,7%
kipas tombol telu						

Pada Tabel 4.11 menunjukkan bahwa akurasi benar untuk semua perintah kipas meningkat seiring dengan bertambahnya *epoch*. Namun pada *epoch 15 early stopping* untuk akurasi benar perintah “mati” dan “*off*” justru menurun dibandingkan dengan *epoch 15* karena model YAMNet terlalu terlatih. Pada pola – pola kipas angin yang terdapat dalam dataset. Model YAMNet menjadi terlalu *sensitive* terhadap pola – pola tersebut, sehingga menjadi lebih sulit mengklasifikasikan suara kipas angin yang berbeda dari pola – pola tersebut.

Tetapi pada perintah suara kipas angin yang mati dan *off* tidak selalu memiliki pola yang sama. Suara kipas angin yang mati dan *off* dapat dipengaruhi oleh berbagai faktor, seperti jenis kipas angin, ukuran kipas angin, dan kondisi lingkungan. Oleh karena itu, model YAMNet yang terlalu terlatih pada pola – pola kipas angin yang terdapat dalam dataset akan lebih sulit untuk mengklasifikasikan suara kipas angin yang mati dan *off* yang tidak memiliki pola yang sama dengan pola – pola yang terdapat dalam dataset.

## 4.5 Pengukuran Tegangan Kapasitor



Gambar 4.22 Pengukuran Tegangan Kapasitor (a) *level* rendah (b) *level* sedang (c) *level* tinggi

Gambar 4.22 menunjukkan hasil pengukuran tegangan pada rangkaian relay kapasitor dengan tiga ukuran yang berbeda yang kemudian diserikan dengan kapasitor. Pada kapasitor *level* rendah (a), perhitungan menghasilkan tegangan sebesar 1120. Namun, saat diuji menggunakan avometer pada kapasitor  $1,2 \mu\text{F}$ , tegangan yang terukur adalah 112V. Ketika kapasitor  $1\mu\text{F}$  digabungkan, tegangan berubah menjadi 104V. Pada kapasitor kedua(b), perhitungan menghasilkan tegangan 147V, tetapi hasil pengukuran pada kapasitor  $2 \mu\text{F}$  menunjukkan tegangan sebesar 84,4V. Ketika digabungkan dengan kapasitor  $1\mu\text{F}$ , tegangan bertambah menjadi 131V. Pada kapasitor ketiga(c), perhitungan menunjukkan tegangan 159V, namun saat diukur pada kapasitor  $2,5 \mu\text{F}$  menggunakan avometer, tegangan yang tercatat 71,5 V. Ketika digabungkan dengan kapasitor  $1\mu\text{F}$ , tegangan meningkat menjadi 143V. Pengukuran ini dilakukan dengan mempertimbangkan bahwa tombol fisik kipas angin berada dalam posisi rendah, yang dapat memengaruhi tingkat tegangan pada kapasitor. Hasil pengujian menunjukkan adanya perbedaan perhitungan teoritis dan pengukuran praktik, yang dapat disebabkan oleh faktor – faktor fisik pada kondisi pengujian, seperti resistansi internal komponen atau kondisi kapasitor yang sebenarnya. Dengan demikian, hasil eksperimen memberikan informasi penting untuk pemahaman yang lebih terhadap respon sistem dalam situasi nyata.

Berikut ini merupakan perhitungan untuk mencari persentase tegangan dari hasil pengukuran melalui perhitungan teori dan Avometer seperti pada persamaan (4.5).

$$\%error = \frac{|Avo-Teori|}{Teori} \times 100\% \quad (4.5)$$

Tabel 4.12 Pengukuran Tegangan Kapasitor Setelah Relay

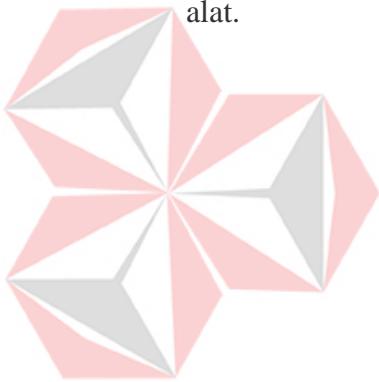
NO	Kapasitor	Hasil Eksperimen		Persentase
		Avometer	Teori	
1	X <sub>C1</sub>	112,2 V	100 V	12,2 %
2	X <sub>C2</sub>	84,4 V	73 V	15 %
3	X <sub>C3</sub>	71,5 V	61 V	17 %

Hasil pengukuran tegangan kapasitor setelah relay dengan menggunakan Avometer menunjukkan perbedaan yang signifikan dari teori perhitungan. Pada kapasitor X<sub>C1</sub> tegangan yang diukur adalah 112,2 V, sedangkan hasil perhitungan teoritisnya seharusnya sebesar 100 V. Hal ini menunjukkan persentase sebesar 12,2% antara hasil pengukuran dan perhitungan teori. Hal serupa terjadi pada kapasitor X<sub>C2</sub>, dimana tegangan yang diukur adalah 84,4 V, sementara hasil perhitungan teoritisnya adalah 73 V, menghasilkan persentase sebesar 15%. Begitu pula pada kapasitor X<sub>C3</sub> dimana tegangan yang diukur adalah 71,5 V, sedangkan hasil perhitungan tegangan teoritisnya hanya 61 V, menunjukkan persentase sebesar 17%. Perbedaan pengukuran ini mengindikasikan adanya faktor – faktor yang memengaruhi nilai tegangan yang diukur, seperti ketidaksempurnaan rangkaian alat yang menyebabkan hasil yang berbeda dari yang diharapkan secara teori.

Tabel 4.13 Pengukuran Tegangan Kapasitor Seri

NO	Kapasitor	Hasil Pengukuran		Persentase
		Avo / Voltmeter	Teori	
1	X <sub>CSERI1</sub>	104V	120 V	13 %
2	X <sub>CSERI2</sub>	131 V	146 V	10,3 %
3	X <sub>CSERI3</sub>	143 V	159 V	10,1%

Hasil persentase pengukuran tegangan kapasitor dalam rangkaian seri antara kapasitor setelah relay dan kapasitor 1 $\mu$ F atau kapasitor output menunjukkan beberapa perbedaan antara nilai yang diukur dan nilai yang dihitung secara teoritis. Pada kapasitor Xc1 hasil pengukuran voltmeter menunjukkan tegangan sebesar 104 V, sementara hasil teori seharusnya sekitar 120 V. Perbedaan antara kedua nilai ini menghasilkan persentase sebesar 13%. Demikian pula, untuk kapasitor Xc2, tegangan yang diukur adalah 131 V, sementara nilai yang didapatkan berdasarkan perhitungan teoritis adalah 146 V, dengan persentase sebesar 10,3%. Begitu juga, untuk kapasitor Xc3, tegangan yang diukur adalah 143 V, sementara nilai yang diperoleh berdasarkan perhitungan teoritis adalah 159 V, dengan persentase sebesar 10,1%. perbedaan antara hasil pengukuran dan perhitungan tersebut menunjukkan kemungkinan adanya faktor – faktor tertentu yang dapat mempengaruhi nilai tegangan yang diukur, seperti ketidaksempurnaan rangkaian alat.



UNIVERSITAS  
Dinamika



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan penelitian yang telah dilakukan pada Bab 4 maka didapatkan kesimpulan sebagai berikut:

1. Proses training dengan 12 *epoch* mempunyai akurasi 89%, *loss* 0,71, *ROC curve class* 0 (kipas angin mati) sebesar 97%, *class* 1 (kipas angin putaran rendah) sebesar 100%, *class* 2 (kipas angin putaran sedang) sebesar 95%, dan *class* 3 (kipas angin putaran tinggi) sebesar 99% .
2. Pada proses *training* dengan 15 *epoch* mempunyai akurasi 100%, *loss* sebesar 0,46, *ROC curve class* 0 (kipas angin mati) sebesar 100%, *class* 1 (kipas angin putaran rendah) sebesar 100%, *class* 2 (kipas angin putaran sedang) sebesar 99%, dan *class* 3 (kipas angin putaran tinggi) sebesar 100% .
3. Dari ke tiga proses training memiliki akurasi yang berbeda. hasil training dataset pertama menggunakan *epoch* 12 mempunyai akurasi 89%, hasil training dataset kedua menggunakan *epoch* 15 mempunyai akurasi 100%, dan hasil training dataset ketiga menggunakan *epoch early stopping* mempunyai akurasi 100% .
4. Proses training dengan 12 *epoch* dengan status:
  - a. Kipas mati menghasilkan *classification report* pada nilai *precision* sebesar 88%, pada nilai *recall* sebesar 80%, dan pada nilai *f1-score* sebesar 84%.
  - b. Kipas angin putaran rendah menghasilkan *classification report* pada nilai *precision* sebesar 90%, pada nilai *recall* sebesar 97%, dan pada nilai *f1-score* sebesar 94%.
  - c. Kipas angin putaran sedang menghasilkan *classification report* pada nilai *precision* sebesar 76%, pada nilai *recall* sebesar 88%, dan pada nilai *f1-score* sebesar 81%.
  - d. Kipas angin putaran tinggi menghasilkan *classification report* pada nilai *precision* sebesar 98%, pada nilai *recall* sebesar 88%, dan pada nilai *f1-score* sebesar 93%.
5. Proses training dengan 15 *epoch* dengan status:

- a. Kipas mati menghasilkan *classification report* pada nilai *precision* sebesar 94%, pada nilai *recall* sebesar 97%, dan pada nilai *f1-score* sebesar 96%.
  - b. Kipas angin putaran rendah menghasilkan *classification report* pada nilai *precision* sebesar 97%, pada nilai *recall* sebesar 100%, dan pada nilai *f1-score* sebesar 99%.
  - c. Kipas angin putaran sedang menghasilkan *classification report* pada nilai *precision* sebesar 100%, pada nilai *recall* sebesar 94%, dan pada nilai *f1-score* sebesar 97%.
  - d. Kipas angin putaran tinggi menghasilkan *classification report* pada nilai *precision* sebesar 100%, pada nilai *recall* sebesar 100%, dan pada nilai *f1-score* sebesar 100%.
6. Proses testing menggunakan model 12 *epoch* dengan perintah secara langsung (*live*) melalui mikrofon laptop dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 10% dan akurasi salah 90%, kipas angin putaran rendah (1) mempunyai akurasi benar 30% dan akurasi salah 70% , kipas angin putaran sedang (2) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, sedangkan kipas angin putaran tinggi (3) mempunyai akurasi benar 20% dan akurasi salah 80%.
  7. Proses testing menggunakan model 15 *epoch* dengan perintah secara langsung (*live*) melalui mikrofon laptop dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 53,3% dan akurasi salah 46%, kipas angin putaran rendah (1) mempunyai akurasi benar 43,3% dan akurasi salah 56,7% , kipas angin putaran sedang (2) mempunyai akurasi benar 36,7% dan akurasi salah 63,3%, sedangkan kipas angin putaran tinggi (3) mempunyai akurasi benar 43,7% dan akurasi salah 56,7%.
  8. Proses testing menggunakan model 15 *epoch early stopping* dengan perintah secara langsung (*live*) melalui mikrofon laptop dari 3 bahasa dengan status kipas mati (0) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, kipas angin putaran rendah (1) mempunyai akurasi benar 66,7% dan akurasi salah 33,3%, kipas angin putaran sedang (2) mempunyai akurasi benar 56,7% dan akurasi salah 43,3%, kipas angin putaran tinggi (3) mempunyai akurasi benar 50% dan akurasi salah 50%.

9. Proses *testing* model dataset subset *test* hasil dari file rekaman audio menggunakan 12 *epoch* yang dilakukan sebanyak 40 percobaan pada semua perintah menghasilkan akurasi sebesar 92,5%.
10. Proses *testing* model dataset subset test menggunakan 15 *epoch* yang dilakukan sebanyak 40 percobaan pada semua perintah memiliki akurasi sebesar 97,5%.
11. Proses *testing* model dataset subses test menggunakan 15 *epoch early stopping* yang dilakukan sebanyak 40 percobaan pada semua perintah memiliki akurasi sebesar 100%.

Kesimpulan akhir dari penelitian ini menunjukkan variasi hasil yang kurang konsisten antara tiga model yang diuji, terutama dalam pengujian klasifikasi suara pengguna secara langsung (*live*).



## 5.2 Saran

Saran untuk penelitian selanjutnya dalam rangka meningkatkan nilai persentase akurasi dan loss dengan memperhatikan kualitas pengambilan dataset audio yang digunakan untuk proses training dengan baik dan standart yang tetap, juga perlu diperhatikan kualitas dari mikrofon yang digunakan. Dapat dipertimbangkan menggunakan metode yang berbeda, salah satunya menggunakan metode klasifikasi suara pada MediaPipe dikarenakan pada saat penelitian ini dilakukan fitur klasifikasi suara pada MediaPipe belum ada dokumentasi atau jurnal yang membahas klasifikasi suara multi – bahasa.

## DAFTAR PUSTAKA

Fahkrudin, Muhammad Aldi (2023). Tugas Akhir: *Sistem Deteksi Gestur Jari Tangan Menggunakan Mediapipe dan Faster-RCNN untuk Mengontrol Kecepatan Kipas Angin*. Repository Universitas Dinamika.

Edowai, Yumerius R.B (2023). Tugas Akhir: *Sistem Automatic Feature Selection Berbasis Deteksi Gestur Kedua Jari Tangan untuk Mengontrol Level Kecepatan Putaran 2 Kipas Angin Menggunakan Mediapipe*. Repository Universitas Dinamika.

Wakerkwa, Fredi (2023). Tugas Akhir: *Kontrol Level Kecepatan Putaran Kipas Angin Melalui Deteksi Bentuk Gestur Jari Tangan Berbasis IoT*. Repository Universitas Dinamika

Firmansyah, Anas Ardi, M. Nanak Zakaria, M. Taufik (2016). *Rancang Bangun Alat Bantu Penyandang Disabilitas Tangan untuk Menghidupkan dan Mematikan Perangkat Elektronik Menggunakan Voice Recognition Module V3*. Jurnal Jaringan Telekomunikasi Politeknik Negeri Malang

Sherila, Ayu Mika (2012). Tugas Akhir: *Sistem Identifikasi Dialek Suku Bangsa di Indonesia Menggunakan Metode Mel – Frequency Cepstral Coefficient dan Self Organizing Map*. Repository Universitas Telkom Bandung

Lestari, Novi, Nelly Khairani Daulay, Armanto (2020). *Simulasi Monitoring Pengatur Kecepatan Kipas Angin Menggunakan Sistem Fuzzy Berbasis Web*. Jurnal Informatika & Rekaya Elektronika

Malmberg, Christopper (2021). *Real – time Audio Classification on an Edge Device Using YAMNet and TensorFlow Lite*. Engineering Degree Project

Ramadhan, Nur Ariesanto, Devi Adi Nufriana (2019). *Rancang Bangun dan Implementasi Sistem Informasi Skripsi Online Berbasis Web*. Jurnal Ilmiah Intech : Information Technology Journal of UMUS

Riberu, F.X Lorens(2023). Tugas Akhir: *Sistem Deteksi Simbol SIBI (SISTEM ISYARAT BAHASA INDONESIA) Secara Realtime Menggunakan MediaPipe Dan LSTM*. Repository Universitas Dinamika.

Wicaksono, Henry, Liliana, Alvin Nathaniel Tjondroviguno (2022). *Pemodelan Lip Reading Bahasa Indonesia Berbasis Visem Menggunakan VGG16 serta Jaro – Winkler Similarity dan Bigram*. Jurnal INFRA

Irwanto, M. Sofyan, Fitra A. Bachtiar, Novanto Yudistira (2022). *Klasifikasi Aktivitas Manusia Menggunakan Algoritme Computed Input Weight Extreme Learning Machine Dengan Reduksi Dimensi Principal Component Analysis*. Jurnal Teknologi Informasi dan Komputer

Fadilah, Zahra Rizky Fadilah, Arie Wahyu Wijayanto (2023). *Perbandingan Motode Klasterisasi Data Bertipe Campuran: One – Hot Encoding, Gower Distance, dan K – Prototype Berdasarkan Akurasi (Studi Kasus: Chronic Kidney Disease Dataset)*. Journal of Applied Informatics and Computing

Puteri, Dian Islamiaty (2023). Implementasi *Long Short Term Memory (LSTM)* dan *Bidirectional Long Short Term Memory (BiLSTM)* Dalam Prediksi Harga Saham Syariah. EULER : Jurnal Ilmiah Matematika, Sains dan Teknologi

Chen, X., Wang, H., & Liu, J. (2023). Bidirectional LSTM for Natural Language Processing. *ACM Computing Surveys*, 56(4), 1-42.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer.



UNIVERSITAS  
Dinamika