



Nama: **Martua Kevin, Dyo Dwi Carol, Rafelina Octa**
Mata Kuliah: **Sistem/Teknologi Multimedia (IF4021)**

Tugas: **Final Project**
Tanggal: 30 Mei 2025

1 Deskripsi Projek

Proyek ini menggabungkan konsep permainan interaktif dengan teknologi pengenalan suara dan visual real-time. Dalam filter ini, pengguna ditantang untuk menirukan suara hewan yang ditampilkan di layar, seperti anjing, kambing, atau kucing. Filter ini memanfaatkan mikrofon untuk merekam suara pengguna selama beberapa detik. Data suara yang direkam dianalisis menggunakan metode RMS untuk mengukur tingkat kekerasan suara, serta FFT (Fast Fourier Transform) untuk mengenali frekuensi dominan. Berdasarkan frekuensi tersebut, sistem menentukan jenis hewan yang ditiru. Jika suara pengguna berhasil dikenali, filter akan menampilkan gambar overlay dari hewan yang sesuai di layar secara real-time menggunakan teknologi OpenCV. Proyek ini menghadirkan pengalaman bermain yang menyenangkan dan edukatif dengan menggabungkan elemen suara, visual, dan interaksi pengguna secara langsung melalui kamera.

2 Teknologi

Teknologi yang digunakan dalam mengerjakan Final Project ini, yaitu sebagai berikut:

- **Bahasa Pemrograman Python**
Python adalah bahasa pemrograman tingkat tinggi yang banyak digunakan dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan, dan pemrosesan sinyal. Python memiliki sintaks yang mudah dipahami, sehingga cocok untuk prototipe cepat dan aplikasi interaktif seperti proyek ini.
- **PyAudio**
PyAudio adalah pustaka Python yang menyediakan antarmuka dengan PortAudio, sebuah pustaka lintas platform untuk menangani input dan output audio. Dalam proyek ini, PyAudio digunakan untuk merekam suara pengguna melalui mikrofon secara real-time.
- **NumPy**
NumPy adalah pustaka Python yang mendukung komputasi numerik tingkat tinggi. Dalam proyek ini, NumPy digunakan untuk mengolah data audio, menghitung RMS (Root Mean Square), dan menerapkan FFT (Fast Fourier Transform) untuk analisis frekuensi suara.
- **OpenCV (cv2)**
OpenCV (Open Source Computer Vision Library) adalah pustaka open-source untuk pemrosesan citra dan pengenalan visual. Dalam proyek ini, OpenCV digunakan untuk menangkap video dari webcam dan menampilkan overlay gambar hewan secara real-time di atas tampilan kamera.
- **FFT (Fast Fourier Transform)**
FFT adalah algoritma untuk mengubah sinyal dari domain waktu ke domain frekuensi. Dalam proyek ini, FFT digunakan untuk menganalisis frekuensi dominan dari suara yang direkam, yang kemudian digunakan untuk mengklasifikasikan jenis suara hewan.

3 Cara Kerja

Cara kerja pada proyek Tirukan Aku adalah sebagai berikut:

- Sistem menggunakan kamera webcam untuk menampilkan gambar tiga hewan (anjing, kambing, dan kucing) yang harus ditirukan suaranya oleh pengguna.
- Setelah pengguna menekan tombol 's' (start), sistem akan mulai merekam suara selama 3 detik menggunakan mikrofon melalui pustaka PyAudio.
- Data audio yang direkam kemudian diolah menggunakan NumPy untuk menghitung RMS (Root Mean Square), guna memastikan suara pengguna cukup keras dan layak dianalisis.
- Jika volume suara memenuhi ambang batas, sistem melanjutkan proses analisis menggunakan FFT (Fast Fourier Transform) untuk mengetahui frekuensi dominan dari suara yang ditirukan pengguna.
- Berdasarkan frekuensi dominan, sistem mengklasifikasikan jenis suara hewan yang ditiru:
 - Frekuensi < 300 Hz \rightarrow Anjing
 - Frekuensi < 1000 Hz \rightarrow Kambing
 - Frekuensi ≥ 1000 Hz \rightarrow Kucing
- Setelah suara dikenali, sistem menggunakan OpenCV untuk menampilkan overlay gambar hewan yang sesuai di atas video dari kamera secara real-time, sebagai bentuk respon visual bahwa suara pengguna berhasil dikenali.
- Pengguna dapat menekan tombol 'q' kapan saja untuk keluar dari tampilan kamera atau menghentikan proses.

4 Penjelasan Kode Program

Kode program dibuat dalam satu file Python dan berfungsi untuk mendeteksi suara hewan (anjing, kambing, dan kucing) berdasarkan frekuensi suara menggunakan mikrofon, serta menampilkan gambar overlay hewan yang sesuai di atas tampilan kamera. Penjelasan ini akan membahas setiap bagian kode secara rinci.

4.1 Import dan Konfigurasi

```
1 import cv2
2 import pyaudio
3 import numpy as np
4 import time
```

- cv2 (OpenCV): Digunakan untuk menangkap video dari kamera dan menampilkan gambar.
- pyaudio: Digunakan untuk menangkap input audio dari mikrofon secara real-time.
- numpy: Digunakan untuk melakukan perhitungan matematis, termasuk Fast Fourier Transform (FFT).
- time: Digunakan untuk mengatur durasi rekaman dan mengukur waktu.

```

1  # Pengaturan Audio untuk PyAudio
2  FORMAT = pyaudio.paInt16      # Format audio 16-bit integer
3  CHANNELS = 1                  # Mono channel
4  RATE = 44100                  # Sample rate 44.1 KHz (CD quality)
5  CHUNK = 1024                  # Jumlah sample per buffer
6  RECORD_SECONDS = 3            # Durasi rekam audio dan video
7
8  # Rentang frekuensi untuk klasifikasi suara hewan
9  FREQ_ANJING_MAX = 300         # Batas maksimum frekuensi anjing (Hz)
10 FREQ_KAMBING_MAX = 1000        # Batas maksimum frekuensi kambing (Hz)
11 MIN_RMS_THRESHOLD = 700       # Batas minimum volume suara yang diterima

```

- **FORMAT:** Menentukan format audio sebagai 16-bit integer.
- **CHANNELS:** Saluran audio mono.
- **RATE:** Laju pengambilan sampel audio sebesar 44.100 Hz.
- **CHUNK:** Jumlah sampel yang diambil per buffer.
- **RECORD_SECONDS:** Durasi rekaman yang digunakan untuk menganalisis suara.
- **FREQ_ANJING_MAX, FREQ_KAMBING_MAX:** Digunakan sebagai batas klasifikasi frekuensi suara.
- **MIN_RMS_THRESHOLD:** Digunakan untuk menentukan ambang minimum kekuatan suara agar dianggap valid.

4.2 Fungsi `calculate_rms(data)`

```

1  def calculate_rms(data):
2      """
3      Menghitung Root Mean Square (RMS) untuk mengukur volume suara.
4      RMS tinggi = suara keras, RMS rendah = suara pelan
5      """
6      data_float = data.astype(np.float32)
7      return np.sqrt(np.mean(data_float**2))

```

Fungsi ini menghitung Root Mean Square (RMS) dari data audio. RMS digunakan untuk mengukur seberapa kuat suara yang diterima. Semakin besar nilai RMS, semakin keras suara yang ditangkap. Nilai ini digunakan untuk menyaring suara yang terlalu pelan agar tidak diproses.

4.3 Fungsi `tampil_awal_dengan_gambar(cap)`

```

1  def tampil_awal_dengan_gambar(cap):
2      """Tampilkan gambar hewan dan kamera. Tekan 's'=start, 'q'=keluar."""
3      ukuran_gambar = (150, 150)
4      anjing = cv2.resize(cv2.imread("Hewan/gambar/anjing.png"), ukuran_gambar)
5      kambing = cv2.resize(cv2.imread("Hewan/gambar/kambing.png"), ukuran_gambar)
6      kucing = cv2.resize(cv2.imread("Hewan/gambar/kucing.png"), ukuran_gambar)
7      barisan_gambar = np.hstack((anjing, kambing, kucing))
8
9      while True:
10         ret, frame = cap.read()
11         if not ret:
12             print("Gagal membaca kamera.")
13             return "exit"
14
15         frame_resized = cv2.resize(frame, (barisan_gambar.shape[1], 300))
16         tampilan = np.vstack((barisan_gambar, frame_resized))

```

```

17     cv2.imshow("Tirukan Suara Hewan ini!!", tampilan)
18
19     key = cv2.waitKey(1) & 0xFF
20     if key == ord('s'):
21         return "start"
22     elif key == ord('q'):
23         return "exit"

```

Fungsi ini bertugas menampilkan antarmuka awal kepada pengguna sebelum proses rekaman suara dilakukan. Tampilan terdiri dari kombinasi gambar hewan dan hasil tangkapan kamera secara real-time.

1. Ukuran Gambar:

- `ukuran_gambar = (150, 150)`: Menentukan resolusi gambar hewan yang akan ditampilkan.

2. Membaca dan Mengubah Ukuran Gambar:

- `cv2.imread(...)`: Membaca gambar hewan dari direktori.
- `cv2.resize(...)`: Mengubah ukuran gambar ke 150x150 piksel agar seragam.
- `np.hstack(...)`: Menyatukan ketiga gambar secara horizontal untuk ditampilkan sebagai satu baris.

3. Loop Tampilan:

- `cap.read()`: Mengambil frame (gambar) dari kamera.
- Jika gagal membaca kamera (not ret), maka keluar dan cetak pesan kesalahan.
- `cv2.resize(frame, ...)`: Mengubah ukuran frame kamera agar sejajar dengan panjang barisan gambar.
- `np.vstack(...)`: Menyatukan barisan gambar (anjing, kambing, kucing) dan frame kamera secara vertikal.
- `cv2.imshow(...)`: Menampilkan gabungan gambar dan kamera dalam satu jendela.

4. Kontrol Keyboard:

- `cv2.waitKey(1) & 0xFF`: Menunggu input dari keyboard.
- `ord('s')`: Jika pengguna menekan huruf 's', maka fungsi mengembalikan string "start" untuk memulai proses rekaman.
- `ord('q')`: Jika pengguna menekan huruf 'q', maka fungsi mengembalikan "exit" untuk keluar dari program.

Fungsi ini memastikan pengguna dapat melihat gambar hewan yang harus ditirukan, serta melihat dirinya sendiri melalui kamera sebelum proses klasifikasi dimulai.

4.4 Fungsi `streaming_kamera_dengan_overlay(...)`

```

1  def streaming_kamera_dengan_overlay(cap, gambar_hewan, hewan_terdeteksi, ukuran_overlay
    =(200,200)):
2      """Streaming kamera dengan overlay gambar hewan."""
3      overlay_resized = cv2.resize(gambar_hewan, ukuran_overlay)
4
5      print(f"Streaming kamera dengan overlay: {hewan_terdeteksi}. Tekan 'q' untuk keluar.")
6      while True:
7          ret, frame = cap.read()
8          if not ret:
9              print("Gagal membaca kamera.")

```

```

10         break
11
12         # Tempel overlay di pojok kiri atas
13         h, w, _ = overlay_resized.shape
14         frame[0:h, 0:w] = overlay_resized
15
16         cv2.imshow("Kamera dengan Overlay Hewan", frame)
17         if cv2.waitKey(1) & 0xFF == ord('q'):
18             break
19
20     cv2.destroyAllWindows()

```

Fungsi ini dijalankan setelah suara hewan berhasil diklasifikasikan. Tujuannya adalah untuk menampilkan gambar hewan yang dikenali di atas video kamera secara langsung.

1. `overlay_resized = cv2.resize(...)`: Gambar hewan diubah ukurannya agar pas sebagai overlay.
2. `cap.read()`: Mengambil frame (gambar) dari kamera secara real-time.
3. Jika frame tidak berhasil diambil, tampilkan pesan kesalahan dan hentikan loop.
4. `frame[0:h, 0:w] = overlay_resized`: Gambar overlay ditempelkan di pojok kiri atas frame video.
5. `cv2.imshow(...)`: Menampilkan hasil frame dengan overlay ke layar.
6. Pengguna dapat keluar dari tampilan dengan menekan tombol q.
7. Setelah keluar dari loop, jendela ditutup dengan `cv2.destroyAllWindows()`.

Fungsi ini memberikan umpan balik visual bahwa suara yang ditirukan telah dikenali dan diproses dengan benar oleh sistem.

4.5 Fungsi `rekam_dan_analisis(cap)`

Fungsi ini merupakan inti dari sistem karena mencakup proses perekaman suara, analisis sinyal audio, serta klasifikasi jenis suara berdasarkan frekuensi.

```

1  def rekam_dan_analisis(cap):
2      # Inisialisasi PyAudio dan buka stream untuk rekaman
3      audio = pyaudio.PyAudio()
4      stream = audio.open(format=FORMAT, channels=CHANNELS,
5                          rate=RATE, input=True,
6                          frames_per_buffer=CHUNK)
7
8      print("Merekam selama 3 detik...")
9      frames_audio = []
10     start_time = time.time()
11
12     while (time.time() - start_time) < RECORD_SECONDS:
13         ret, frame = cap.read()
14         if ret:
15             cv2.imshow('Merekam... Tekan Q untuk stop', frame)
16             try:
17                 data_audio = stream.read(CHUNK, exception_on_overflow=False)
18                 frames_audio.append(data_audio)
19             except:
20                 print("Overflow audio.")
21
22         if cv2.waitKey(1) & 0xFF == ord('q'):
23             print("Rekaman dihentikan.")
24             break

```

```
25
26 stream.stop_stream()
27 stream.close()
28 audio.terminate()
29 cv2.destroyAllWindows()
30
31 if not frames_audio:
32     print("Tidak ada data audio.")
33     return
34
35 audio_np = np.frombuffer(b''.join(frames_audio), dtype=np.int16)
```

1. Inisialisasi PyAudio:

- `pyaudio.PyAudio()`: Membuat objek PyAudio untuk menangani input suara.
- `audio.open(...)`: Membuka stream untuk merekam suara dengan parameter yang telah ditentukan sebelumnya.

2. Mulai Rekaman:

- Rekaman dilakukan selama durasi 3 detik menggunakan loop while.
- `cap.read()`: Mengambil frame dari kamera untuk ditampilkan selama rekaman.
- `stream.read(...)`: Membaca data audio dari mikrofon dalam ukuran buffer CHUNK.
- `frames_audio.append(...)`: Menyimpan potongan-potongan data audio ke dalam list untuk diproses nanti.
- `cv2.imshow(...)`: Menampilkan frame kamera dengan teks "Merekam... Tekan Q untuk stop".

3. Kontrol Keluar:

- Jika pengguna menekan tombol q, maka proses rekaman akan dihentikan lebih awal.

4. Menutup Stream:

- Setelah rekaman selesai atau dihentikan, stream audio ditutup dan koneksi dengan PyAudio dihentikan dengan `terminate()`.
- Semua jendela OpenCV juga ditutup.

5. Validasi Data Audio:

- Jika `frames_audio` kosong, program mencetak peringatan dan tidak melanjutkan proses analisis.

6. Menggabungkan dan Mengubah Data:

- `b''.join(...)`: Menggabungkan semua potongan audio menjadi satu.
- `np.frombuffer(..., dtype=np.int16)`: Mengubah data audio menjadi array NumPy bertipe integer 16-bit yang siap dianalisis lebih lanjut.

```
1 # Tambahkan pengecekan RMS
2 rms_value = calculate_rms(audio_np)
3 print(f"RMS suara yang direkam: {rms_value:.2f}")
4
5 if rms_value < MIN_RMS_THRESHOLD:
6     print(f"Suara terlalu pelan (RMS: {rms_value:.2f}). Coba lebih keras!")
7     return
8
```

```

9     print("Suara cukup keras, melanjutkan analisis frekuensi...")
10
11     # Analisis Frekuensi menggunakan FFT
12     fft_data = np.fft.fft(audio_np)                # Transformasi Fourier
13     freqs = np.fft.fftfreq(len(fft_data), 1.0/RATE) # Konversi ke Hz
14     idx = np.where(freqs > 0)                      # Ambil frekuensi positif
15     fft_positive = np.abs(fft_data[idx])            # Magnitude frekuensi
16     freqs_positive = freqs[idx]

```

7. Perhitungan RMS:

- `calculate_rms(audio_np)`: Fungsi ini digunakan untuk menghitung kekuatan suara dari sinyal audio.
- Jika hasil RMS lebih kecil dari `MIN_RMS_THRESHOLD`, suara dianggap terlalu pelan dan proses dihentikan.

8. Transformasi Fourier dan Analisis Frekuensi:

- `np.fft.fft(...)`: Mengubah sinyal audio dari domain waktu ke domain frekuensi.
- `np.fft.fftfreq(...)`: Menghasilkan array frekuensi yang sesuai dengan hasil FFT.
- `np.where(freqs > 0)`: Mengambil hanya bagian spektrum positif.
- `np.abs(...)`: Mengambil nilai absolut (magnitudo) dari hasil FFT untuk mengetahui kekuatan tiap frekuensi.

```

1     # Deteksi jenis hewan berdasarkan frekuensi dominan
2     dominant_freq = freqs_positive[np.argmax(fft_positive)]
3     if dominant_freq < FREQ_ANJING_MAX:          # Frek < 300 Hz = Anjing
4         hewan_terdeteksi = "Anjing"
5         gambar = cv2.imread("Hewan/gambar/anjing.png")
6     elif dominant_freq < FREQ_KAMBING_MAX:        # Frek < 1000 Hz = Kambing
7         hewan_terdeteksi = "Kambing"
8         gambar = cv2.imread("Hewan/gambar/kambing.png")
9     else:                                         # Frek >= 1000 Hz = Kucing
10        hewan_terdeteksi = "Kucing"
11        gambar = cv2.imread("Hewan/gambar/kucing.png")
12
13    print(f"Hewan yang ditiru: {hewan_terdeteksi}")
14
15    if gambar is not None:
16        streaming_kamera_dengan_overlay(cap, gambar, hewan_terdeteksi, ukuran_overlay=(200,
17        200))

```

9. Klasifikasi Hewan berdasarkan Frekuensi Dominan:

- `np.argmax(...)`: Digunakan untuk mencari indeks frekuensi dominan (frekuensi paling kuat).
- Frekuensi tersebut kemudian dibandingkan dengan batas klasifikasi:
 - Frekuensi < 300 Hz → Anjing
 - Frekuensi < 1000 Hz → Kambing
 - Frekuensi ≥ 1000 Hz → Kucing
- Gambar hewan yang sesuai kemudian dimuat (di-load) ke dalam program.

10. Visualisasi Gambar Hewan:

- Jika gambar ditemukan, fungsi `streaming_kamera_dengan_overlay()` dipanggil untuk menampilkan gambar tersebut di atas tampilan kamera.

4.6 Fungsi main()

Fungsi ini adalah pengatur alur utama dari program:

```
1 def main():
2     cap = cv2.VideoCapture(0)
3     if not cap.isOpened():
4         print("Tidak bisa membuka kamera.")
5         return
6
7     while True:
8         aksi = tampil_awal_dengan_gambar(cap)
9         if aksi == "start":
10             rekam_dan_analisis(cap)
11         elif aksi == "restart":
12             print("Tampilan awal dimuat ulang.")
13             continue
14         elif aksi == "exit":
15             break
16
17     cap.release()
18     cv2.destroyAllWindows()
```

1. Membuka koneksi ke kamera.
2. Menjalankan fungsi `tampil_awal_dengan_gambar()` untuk memulai antarmuka.
3. Berdasarkan input pengguna, program akan menjalankan rekaman suara atau keluar dari aplikasi.
4. Menutup koneksi kamera dan jendela OpenCV setelah program selesai.
5. Program dijalankan dengan baris `if __name__ == '__main__': main()`, yang memastikan bahwa fungsi `main()` hanya dijalankan jika file ini dieksekusi secara langsung, bukan saat diimpor sebagai modul.

5 Hasil dan Pembahasan

hasil dari implementasi program deteksi suara hewan serta bagaimana sistem merespons suara yang diberikan oleh pengguna. Hasil diuji dengan menggunakan tiga jenis suara hewan: anjing, kambing, dan kucing. Masing-masing diuji berdasarkan input audio dan diamati hasil output visual yang ditampilkan pada layar.

5.1 Program Utama

Bagian program utama ini bertanggung jawab untuk menjalankan sistem deteksi suara hewan. Berikut adalah implementasi dari bagian tersebut:

```
1 if __name__ == '__main__':
2     main()
```

5.1.1 Deskripsi Kode

Kode ini melakukan langkah-langkah berikut:

1. Memeriksa Nama Modul: Kode dimulai dengan pernyataan `if __name__ == '__main__'` yang memeriksa apakah skrip dijalankan sebagai program utama. Hal ini penting agar bagian kode tersebut hanya dieksekusi saat file dijalankan secara langsung, bukan saat diimpor sebagai modul oleh program lain.

2. Menjalankan Fungsi Utama: Jika kondisi tersebut terpenuhi, maka fungsi `main()` akan dipanggil. Fungsi `main()` bertanggung jawab atas pengelolaan seluruh alur program, termasuk pembukaan kamera, penampilan menu awal, proses perekaman suara, serta analisis dan visualisasi hasil.
3. Loop Interaktif dalam `main()`: Di dalam fungsi `main()`, terdapat loop `while True` yang menjaga agar program terus berjalan hingga pengguna memilih keluar.
4. Fungsi `tampil_awal_dengan_gambar(cap)` dipanggil untuk menampilkan antarmuka menu awal kepada pengguna. Berdasarkan input pengguna (`start`, `exit`), sistem akan mengeksekusi perekaman suara dan analisis, atau mengakhiri program.

5.2 Hasil Tampilan Awal

Saat program dijalankan, fungsi `tampil_awal_dengan_gambar(cap)` akan menampilkan tiga gambar hewan (anjing, kambing, dan kucing) di bagian atas layar dan menampilkan video real-time dari kamera di bawah gambar. Tampilan ini memberikan petunjuk kepada pengguna untuk meniru salah satu dari ketiga suara hewan tersebut. Pengguna dapat menekan tombol `s` untuk memulai proses atau `q` untuk keluar dari program.

5.3 Hasil Perekaman dan Analisis Suara

- Setelah pengguna menekan `s`, sistem akan melakukan perekaman suara selama 3 detik. Selama proses ini, video dari kamera tetap ditampilkan untuk memastikan pengguna tetap berada dalam frame.
- Hasil perekaman akan diproses menggunakan fungsi `calculate_rms()` untuk memastikan volume suara cukup untuk dianalisis. Jika RMS kurang dari ambang batas, sistem akan menampilkan peringatan bahwa suara terlalu pelan.
- Jika suara lolos ambang batas RMS, maka sistem akan melanjutkan ke proses transformasi Fourier (FFT) untuk mengetahui spektrum frekuensi dari suara yang diterima. Frekuensi dominan dari sinyal audio akan dibandingkan dengan rentang klasifikasi untuk menentukan jenis suara:
 - Frekuensi < 300 Hz \rightarrow Anjing
 - Frekuensi < 1000 Hz \rightarrow Kambing
 - Frekuensi ≥ 1000 Hz \rightarrow Kucing

5.4 Hasil Visualisasi Output

Jika suara dikenali, maka sistem akan menampilkan hasil pengenalan berupa gambar hewan yang sesuai dengan suara yang ditiru. Fungsi `streaming_kamera_dengan_overlay()` digunakan untuk menampilkan gambar hewan di pojok kiri atas layar kamera secara real-time. Pengguna dapat menekan tombol `q` untuk keluar dari tampilan ini dan kembali ke menu awal.

Berikut ini adalah contoh hasil dari sistem:

- Saat suara anjing ditiru, sistem mengenali suara tersebut dan menampilkan gambar anjing.
- Saat suara kambing ditiru, sistem mengenali suara tersebut dan menampilkan gambar kambing.
- Saat suara kucing ditiru, sistem mengenali suara tersebut dan menampilkan gambar kucing.

5.5 Evaluasi dan Pembahasan

Sistem bekerja dengan baik dalam mengenali suara jika:

- Volume suara cukup keras (RMS melebihi ambang minimal).
- Lingkungan relatif tenang dan minim noise. Namun, terdapat beberapa catatan penting:
- Sistem dapat salah mengenali suara jika terdapat gangguan atau kebisingan latar.
- Rentang frekuensi yang digunakan bersifat sederhana dan belum melibatkan teknik machine learning untuk klasifikasi yang lebih kompleks. Sistem bergantung pada kualitas mikrofon dan kemampuan pengguna dalam menirukan suara hewan dengan akurat.

Secara keseluruhan, sistem berhasil menampilkan output visual sesuai dengan suara yang dikenali, sehingga dapat digunakan sebagai media edukatif atau hiburan interaktif.

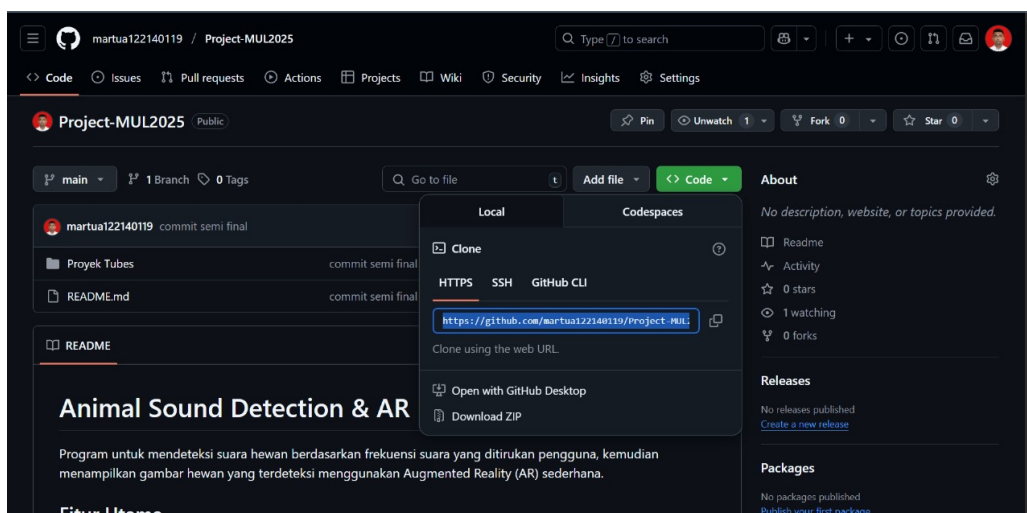
6 Implementasi Program

Bab ini menjelaskan langkah-langkah implementasi dari program "*Tirukan Aku*", mulai dari pengambilan repository hingga menjalankan aplikasi. Setiap langkah didukung dengan panduan visual sesuai gambar yang ada dalam laporan.

6.1 Langkah 1: Menyalin URL Repository

Langkah pertama adalah mendapatkan salinan kode sumber dari GitHub.

1. Buka halaman repository di: <https://github.com/martua122140119/Project-MUL2025>
2. Klik tombol **Code** (berwarna hijau) di bagian atas halaman.
3. Pilih opsi **HTTPS** dan salin URL repository yang muncul.

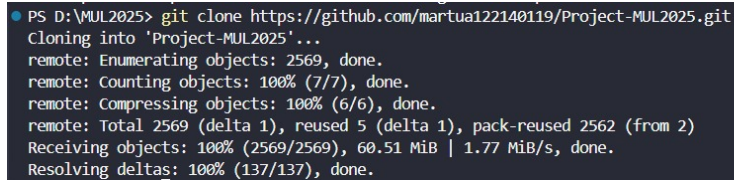


Gambar 1: Salin URL Repository

6.2 Langkah 2: Clone Repository

Setelah URL disalin, langkah berikutnya adalah menyalin isi repository ke komputer lokal menggunakan perintah berikut:

```
git clone https://github.com/martua122140119/Project-MUL2025
```



```
PS D:\MUL2025> git clone https://github.com/martua122140119/Project-MUL2025.git
Cloning into 'Project-MUL2025'...
remote: Enumerating objects: 2569, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 2569 (delta 1), reused 5 (delta 1), pack-reused 2562 (from 2)
Receiving objects: 100% (2569/2569), 60.51 MiB | 1.77 MiB/s, done.
Resolving deltas: 100% (137/137), done.
```

Gambar 2: Git Clone Repository

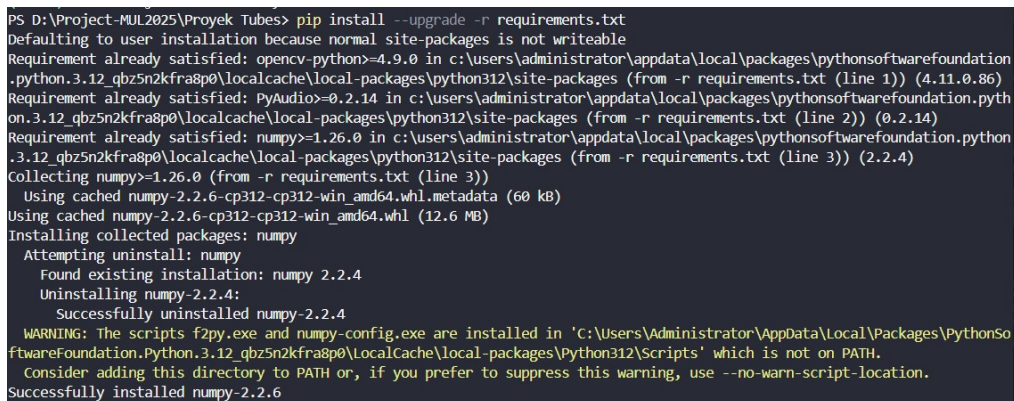
6.3 Langkah 3: Install Dependencies

Untuk menjalankan program, pastikan dependensi berikut telah terinstal:

- **OpenCV** untuk pengolahan video dan tampilan overlay.
- **NumPy** untuk komputasi numerik dan analisis sinyal.
- **PyAudio** untuk perekaman suara dari mikrofon.

Instalasi dilakukan dengan perintah berikut:

```
pip install opencv-python numpy pyaudio
```



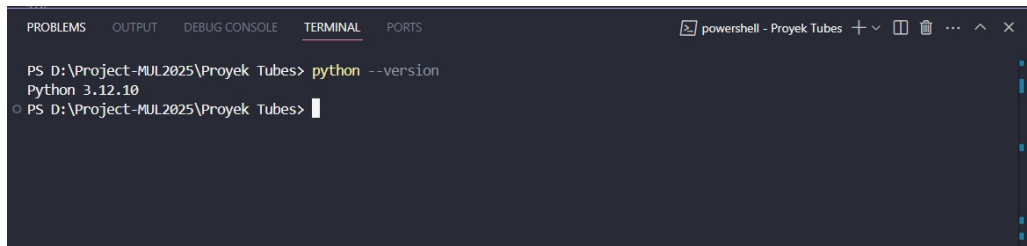
```
PS D:\Project-MUL2025\Proyek Tubes> pip install --upgrade -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: opencv-python>=4.9.0 in c:\users\administrator\appdata\local\packages\pythonsoftwarefoundation.python.3.12.qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from -r requirements.txt (line 1)) (4.11.0.86)
Requirement already satisfied: PyAudio>=0.2.14 in c:\users\administrator\appdata\local\packages\pythonsoftwarefoundation.python.3.12.qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from -r requirements.txt (line 2)) (0.2.14)
Requirement already satisfied: numpy>=1.26.0 in c:\users\administrator\appdata\local\packages\pythonsoftwarefoundation.python.3.12.qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from -r requirements.txt (line 3)) (2.2.4)
Collecting numpy>=1.26.0 (from -r requirements.txt (line 3))
  Using cached numpy-2.2.6-cp312-cp312-win_amd64.whl.metadata (60 kB)
Using cached numpy-2.2.6-cp312-cp312-win_amd64.whl (12.6 MB)
Installing collected packages: numpy
  Attempting uninstall: numpy
    Found existing installation: numpy 2.2.4
    Uninstalling numpy-2.2.4:
      Successfully uninstalled numpy-2.2.4
  WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\Administrator\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12.qbz5n2kfra8p0\LocalCache\local-packages\Python312\Scripts' which is not on PATH.
    Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-2.2.6
```

Gambar 3: Install Dependencies

6.4 Langkah 4: Cek Versi Python

Versi Python minimal yang disarankan adalah Python 3.x. Untuk memastikan versi Python sudah sesuai:

```
python --version
```

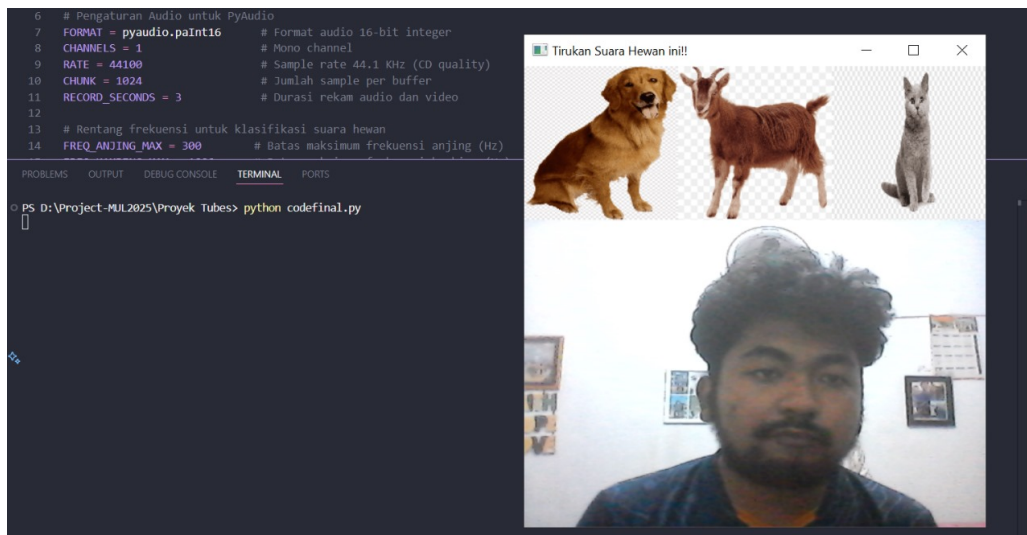


Gambar 4: Cek Versi Python

6.5 Langkah 5: Menjalankan Program

Masuk ke direktori hasil clone dan jalankan program utama.

```
cd Project-MUL2025
python tirukan_aku.py
```

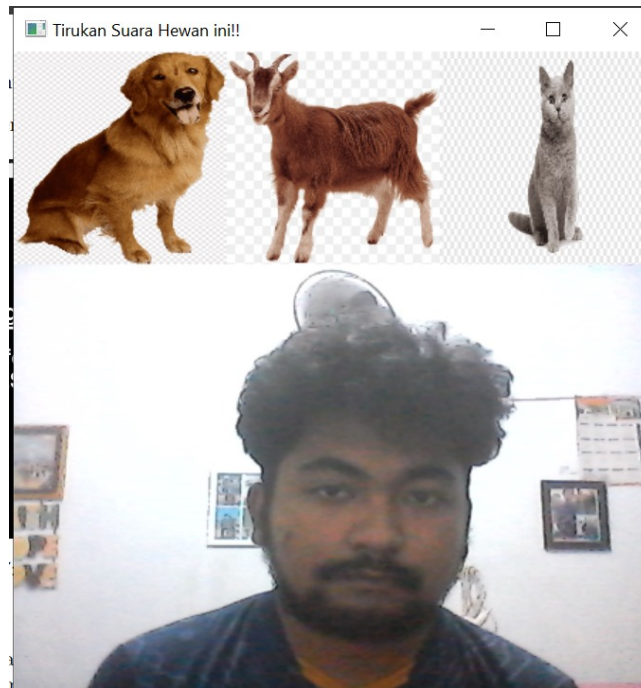


Gambar 5: Menjalankan Kode

6.6 Tampilan Antarmuka Program

Setelah program dijalankan, antarmuka pengguna akan menampilkan elemen visual interaktif untuk memandu pengguna.

Gambar 6: Tampilan Awal Aplikasi Pada tahap ini, program menampilkan tiga gambar hewan (anjing, kambing, kucing) di bagian atas jendela, dan tampilan video dari kamera pengguna di bawahnya. Ini bertujuan memberi arahan kepada pengguna mengenai suara hewan yang dapat ditirukan.



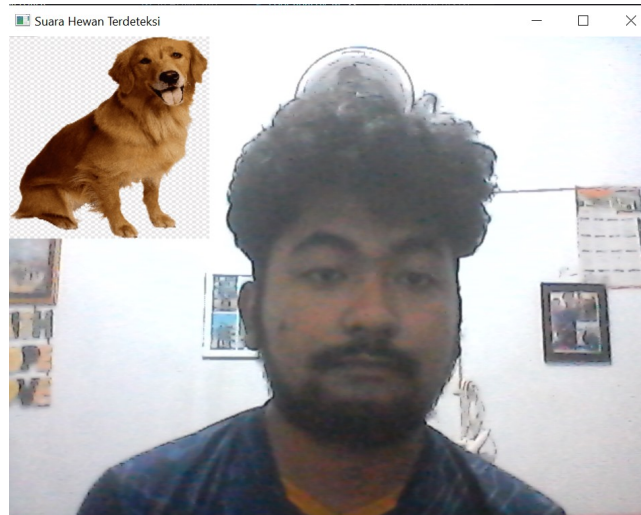
Gambar 6: Tampilan Awal Aplikasi

Gambar 7: Proses Merekam Suara Saat pengguna menekan tombol **s**, program mulai merekam suara selama 3 detik. Selama proses perekaman, video dari kamera tetap ditampilkan agar pengguna tetap berada dalam posisi yang sesuai.



Gambar 7: Proses Merekam Suara

Gambar 8: Suara Hewan Terdeteksi Jika suara yang direkam berhasil dikenali oleh sistem berdasarkan frekuensi dominan, maka program akan menampilkan gambar overlay hewan yang sesuai di pojok layar. Hal ini memberikan umpan balik visual kepada pengguna bahwa suara berhasil diklasifikasi.



Gambar 8: Suara Hewan Terdeteksi

6.7 Kesimpulan

Langkah-langkah implementasi di atas memandu pengguna untuk menjalankan sistem deteksi suara hewan dengan sukses. Program menggabungkan teknologi real-time audio dan visual untuk menghasilkan interaksi yang edukatif dan menarik, khususnya bagi pengguna anak-anak.