



Nama: **Muhammad Daffa Rafif Wibowo (122140036)** Tugas Ke: **Worksheet 1: Setup Python Environment untuk Multimedia**

Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)**

Tanggal: August 29, 2025

1 Tujuan Pembelajaran

Setelah menyelesaikan worksheet ini, mahasiswa diharapkan mampu:

- Memahami pentingnya manajemen environment Python untuk pengembangan multimedia
- Menginstall dan mengkonfigurasi Python environment menggunakan conda, venv, atau uv
- Menginstall library-library Python yang diperlukan untuk multimedia processing
- Memverifikasi instalasi dengan mengimpor dan menguji library multimedia
- Mendokumentasikan proses konfigurasi dan hasil pengujian dalam format $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

2 Latar Belakang

Python telah menjadi bahasa pemrograman yang sangat populer untuk multimedia processing karena memiliki ekosistem library yang sangat kaya. Namun, untuk dapat bekerja dengan multimedia secara efektif, kita perlu mengatur environment Python dengan benar dan menginstall library-library yang tepat.

Manajemen environment Python sangat penting untuk:

- Menghindari konflik antar library (dependency conflict)
- Memastikan reproducibility dari project
- Memudahkan kolaborasi antar developer
- Memisahkan project yang berbeda dengan requirement yang berbeda

3 Instruksi Tugas

3.1 Persiapan

Sebelum memulai, pastikan Anda telah:

- Menginstall Python 3.8 atau lebih baru di sistem Anda
- Memilih salah satu tool manajemen environment: **conda**, **venv**, atau **uv**
- Membuka terminal/command prompt
- Menyiapkan dokumen $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ini untuk dokumentasi

3.2 Bagian 1: Membuat Environment Python

Pilih **SALAH SATU** dari tiga opsi berikut dan ikuti langkah-langkahnya:

3.2.1 Opsi 1: Menggunakan Conda (Direkomendasikan untuk pemula)

Jalankan perintah berikut di terminal:

```
1 # Membuat environment baru dengan nama 'multimedia'
2 conda create -n multimedia python=3.11
3
4 # Mengaktifkan environment
5 conda activate multimedia
6
7 # Verifikasi environment aktif
8 conda info --envs
```

Kode 1: Membuat environment dengan Conda

3.2.2 Opsi 2: Menggunakan venv (Built-in Python)

```
1 # Membuat environment baru
2 python3 -m venv multimedia-env
3
4 # Mengaktifkan environment (Linux/Mac)
5 source multimedia-env/bin/activate
6
7 # Mengaktifkan environment (Windows)
8 # multimedia-env\Scripts\activate
9
10 # Verifikasi environment aktif
11 which python
```

Kode 2: Membuat environment dengan venv

3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
1 # Install uv terlebih dahulu jika belum ada
2 # pip install uv
3
4 # Membuat environment baru
5 uv venv multimedia-uv
6
7 # Mengaktifkan environment (Linux/Mac)
8 source multimedia-uv/bin/activate
9
10 # Mengaktifkan environment (Windows)
11 # multimedia-uv\Scripts\activate
12
13 # Verifikasi environment aktif
14 which python
```

Kode 3: Membuat environment dengan uv

Dokumentasikan di sini:

- Tool manajemen environment yang Anda pilih: [Membuat Enviroment dengan uv]
- Screenshot atau copy-paste output dari perintah verifikasi environment

3.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

3.3.1 Library Audio Processing

```
1 # Untuk conda:
2 conda install -c conda-forge librosa soundfile scipy
3
4 # Untuk pip (venv/uv):
5 pip install librosa soundfile scipy
```

Kode 4: Instalasi library audio

3.3.2 Library Image Processing

```
1 # Untuk conda:
2 conda install -c conda-forge opencv pillow scikit-image matplotlib
3
4 # Untuk pip (venv/uv):
5 pip install opencv-python pillow scikit-image matplotlib
```

Kode 5: Instalasi library image

3.3.3 Library Video Processing

```
1 # Untuk conda:
2 conda install -c conda-forge ffmpeg
3 pip install moviepy
4
5 # Untuk pip (venv/uv):
6 pip install moviepy
```

Kode 6: Instalasi library video

3.3.4 Library General Purpose

```
1 # Untuk conda:
2 conda install numpy pandas jupyter
3
4 # Untuk pip (venv/uv):
5 pip install numpy pandas jupyter
```

Kode 7: Instalasi library umum

Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan
- Screenshot proses instalasi atau output sukses
- Daftar library yang berhasil diinstall dengan versinya

3.4 Bagian 3: Verifikasi Instalasi

Buat file Python sederhana untuk menguji semua library yang telah diinstall:

Jalankan script dan dokumentasikan hasilnya:

3.5 Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

3.5.1 Test Audio Processing

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Generate simple sine wave
5 duration = 2 # seconds
6 sample_rate = 44100
7 frequency = 440 # A4 note
8
9 t = np.linspace(0, duration, int(sample_rate * duration))
10 audio_signal = np.sin(2 * np.pi * frequency * t)
11
12 # Plot waveform
13 plt.figure(figsize=(10, 4))
14 plt.plot(t[:1000], audio_signal[:1000]) # Plot first 1000 samples
15 plt.title('Sine Wave (440 Hz)')
16 plt.xlabel('Time (s)')
17 plt.ylabel('Amplitude')
18 plt.grid(True)
19 plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
20 plt.show()
21
22 print(f"Generated {duration}s sine wave at {frequency}Hz")
23 print(f"Sample rate: {sample_rate}Hz")
24 print(f"Total samples: {len(audio_signal)}")

```

Kode 8: Test audio processing sederhana

3.5.2 Test Image Processing

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5 # Create a simple test image
6 width, height = 400, 300
7 image = np.zeros((height, width, 3), dtype=np.uint8)
8
9 # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 255 # Green section
12 image[:, 2*width//3:, 2] = 255 # Blue section
13
14 # Add a white circle in the center
15 center_x, center_y = width//2, height//2
16 radius = 50
17 Y, X = np.ogrid[:height, :width]
18 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
19 image[mask] = [255, 255, 255]
20
21 # Display and save
22 plt.figure(figsize=(8, 6))
23 plt.imshow(image)
24 plt.title('Test Image with RGB Stripes and White Circle')
25 plt.axis('off')
26 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')

```

```

27 plt.show()
28
29 print(f"Created test image: {width}x{height} pixels")
30 print(f"Image shape: {image.shape}")
31 print(f"Image dtype: {image.dtype}")

```

Kode 9: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas
- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)
- Error message jika ada dan cara mengatasinya

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

Copy-paste output lengkap dari script test_multimedia.py di sini:

```

1 [import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Generate simple sine wave
5 duration = 2 # seconds
6 sample_rate = 44100
7 frequency = 440 # A4 note
8
9 t = np.linspace(0, duration, int(sample_rate * duration))
10 audio_signal = np.sin(2 * np.pi * frequency * t)
11
12 # Plot waveform
13 plt.figure(figsize=(10, 4))
14 plt.plot(t[:1000], audio_signal[:1000]) # Plot first 1000 samples
15 plt.title('Sine Wave (440 Hz)')
16 plt.xlabel('Time (s)')
17 plt.ylabel('Amplitude')
18 plt.grid(True)
19 plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
20 plt.show()
21
22 print(f"Generated {duration}s sine wave at {frequency}Hz")
23 print(f"Sample rate: {sample_rate}Hz")
24 print(f"Total samples: {len(audio_signal)}")
25
26 RGB
27
28 import numpy as np
29 import matplotlib.pyplot as plt
30 from PIL import Image
31
32 # Create a simple test image
33 width, height = 400, 300
34 image = np.zeros((height, width, 3), dtype=np.uint8)
35
36 # Add some patterns
37 image[:, :width//3, 0] = 255 # Red section
38 image[:, width//3:2*width//3, 1] = 255 # Green section
39 image[:, 2*width//3:, 2] = 255 # Blue section
40

```

```

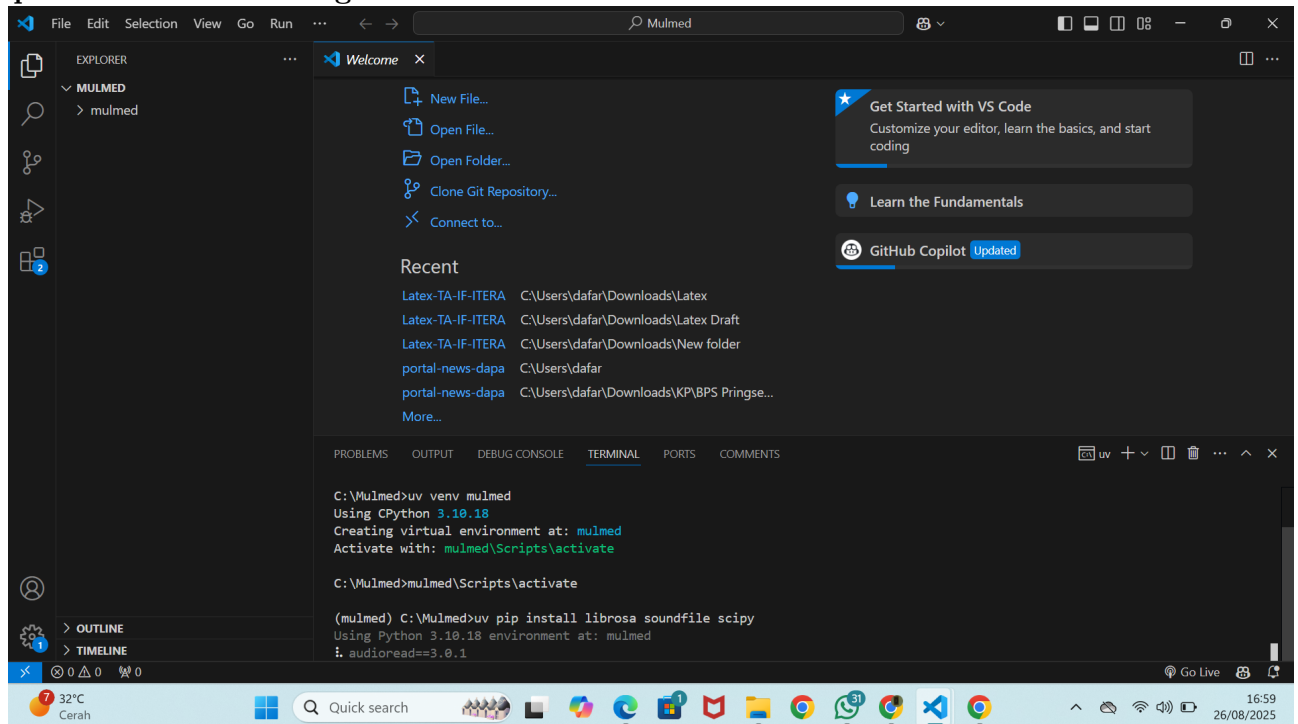
41 # Add a white circle in the center
42 center_x, center_y = width//2, height//2
43 radius = 50
44 Y, X = np.ogrid[:height, :width]
45 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
46 image[mask] = [255, 255, 255]
47
48 # Display and save
49 plt.figure(figsize=(8, 6))
50 plt.imshow(image)
51 plt.title('Test Image with RGB Stripes and White Circle')
52 plt.axis('off')
53 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
54 ]

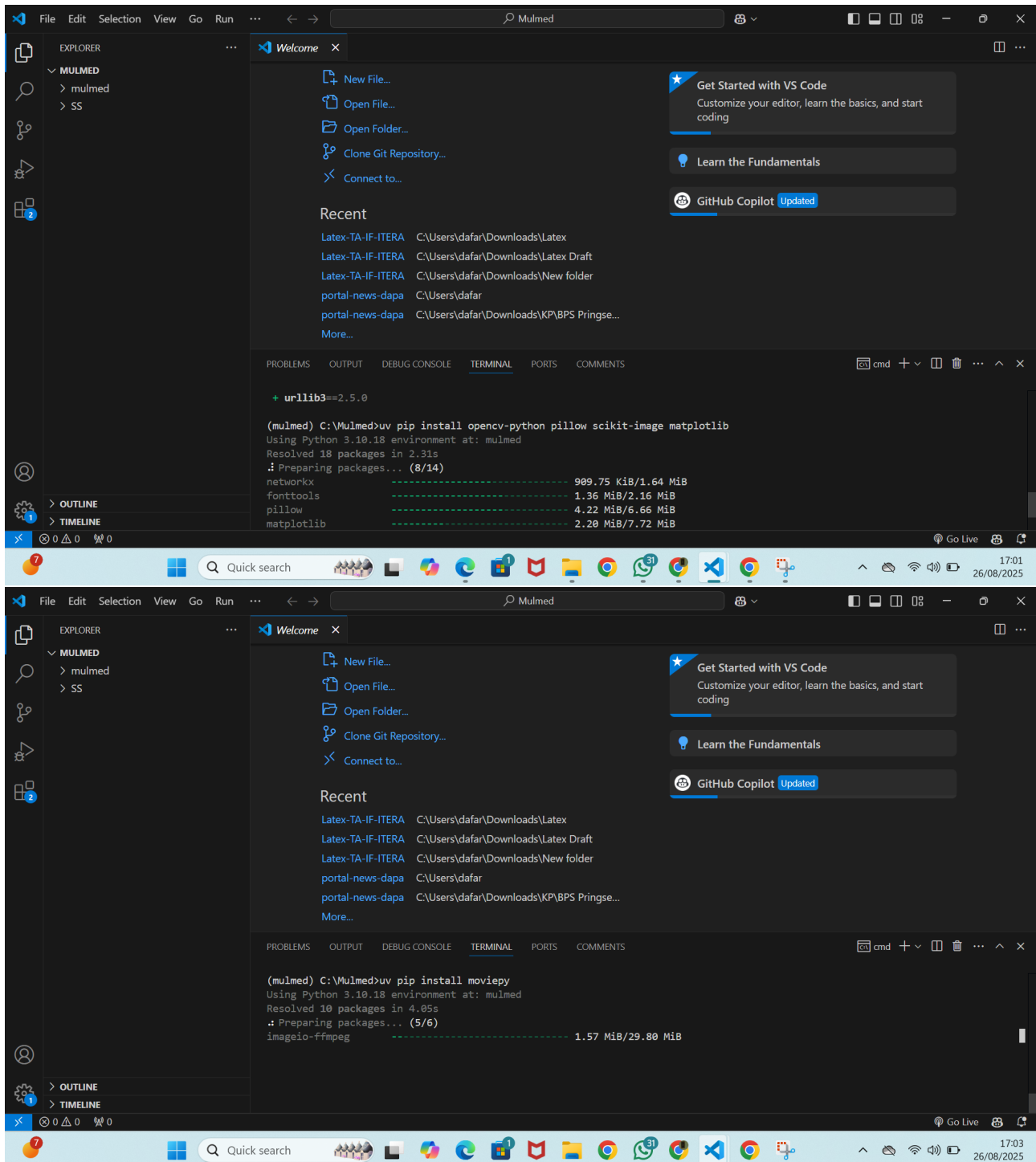
```

Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:





The screenshot displays the Visual Studio Code interface. The top panel shows the 'Welcome' screen with options to create a new file, open a file, or connect to a repository. The bottom panel shows a Jupyter Notebook with a sine wave plot and code for generating and saving the wave.

Terminal Output:

```
(mulmed) C:\Mulmed>uv pip install numpy pandas jupyter
Using Python 3.10.18 environment at: mulmed
+ colorama==0.4.6
+ imageio-ffmpeg==0.6.0
+ moviepy==2.2.1
+ proglog==0.1.12
+ python-dotenv==1.1.1
+ tqdm==4.67.1
```

Jupyter Notebook Code:

```
print(f"Sample rate: {sample_rate}Hz")
print(f"Total samples: {len(audio_signal)}")
```

Plot:

Sine Wave (440 Hz)

Amplitude

Time (s)

Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200

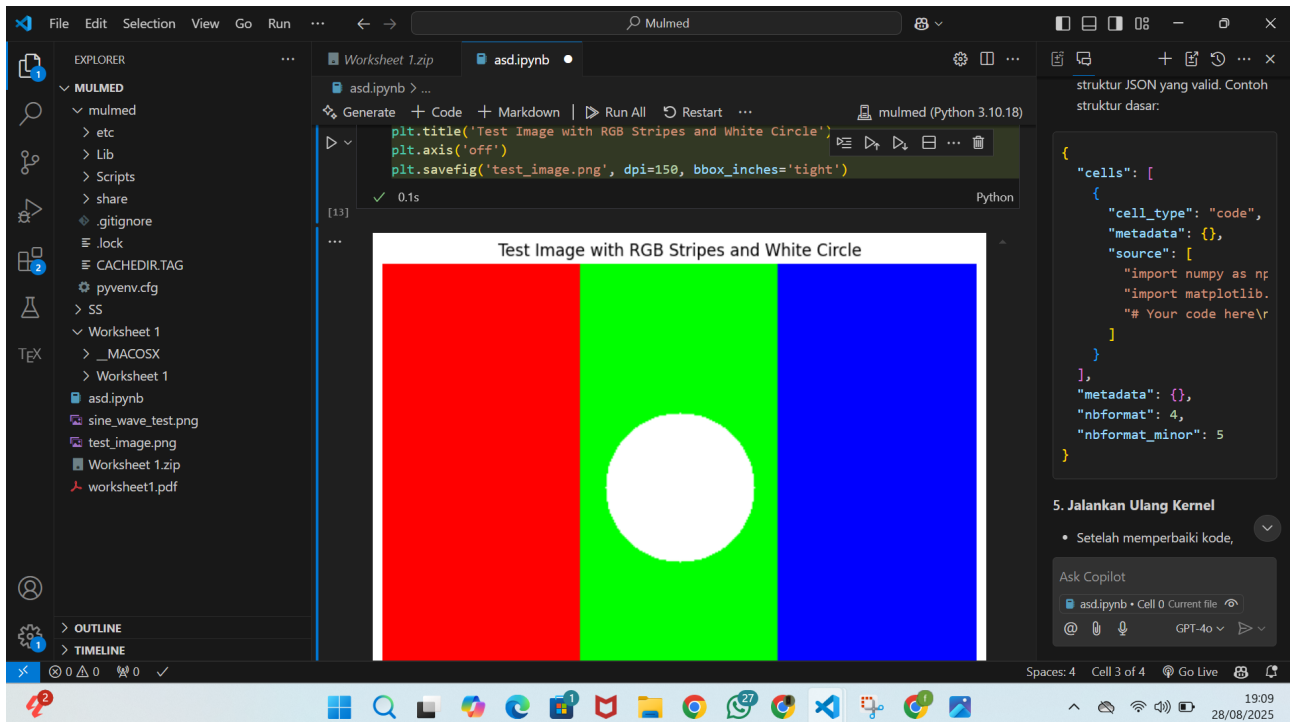
Code Snippet:

```
import numpy as np
import matplotlib.pyplot as plt

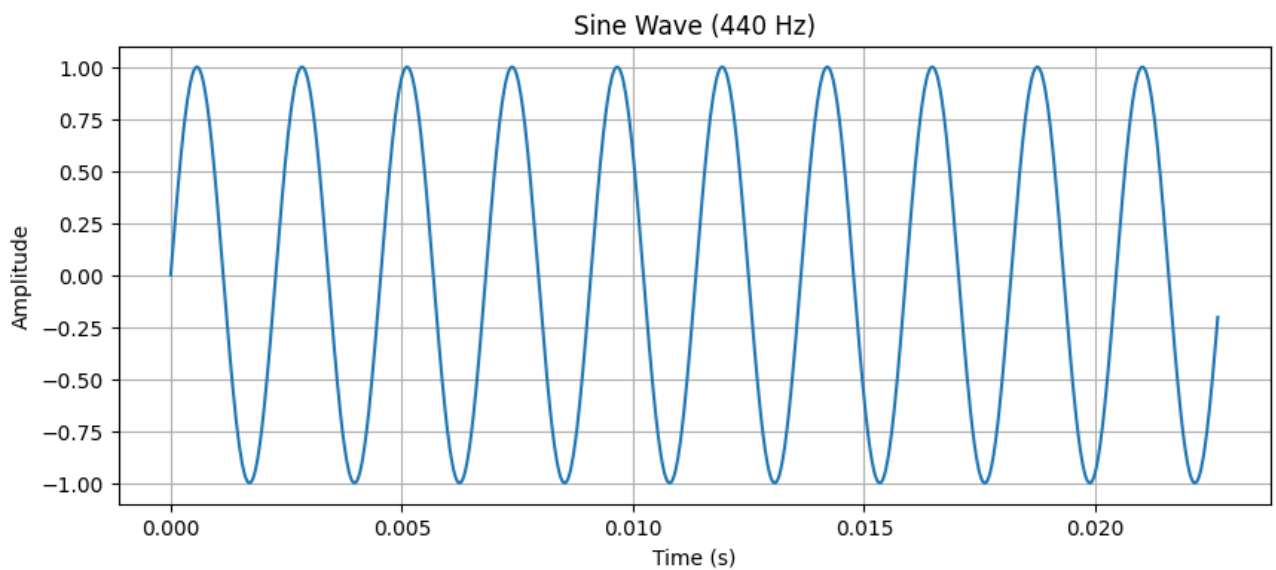
# Generate simple sine wave
duration = 2 # seconds
sample_rate = 44100
frequency = 440 # A4 note

t = np.linspace(0, duration,
audio_signal = np.sin(2 * r

# Plot waveform
plt.figure(figsize=(10, 4))
plt.plot(t[:1000], audio_si
plt.title('Sine Wave (440 t
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.savefig('sine wave test
```

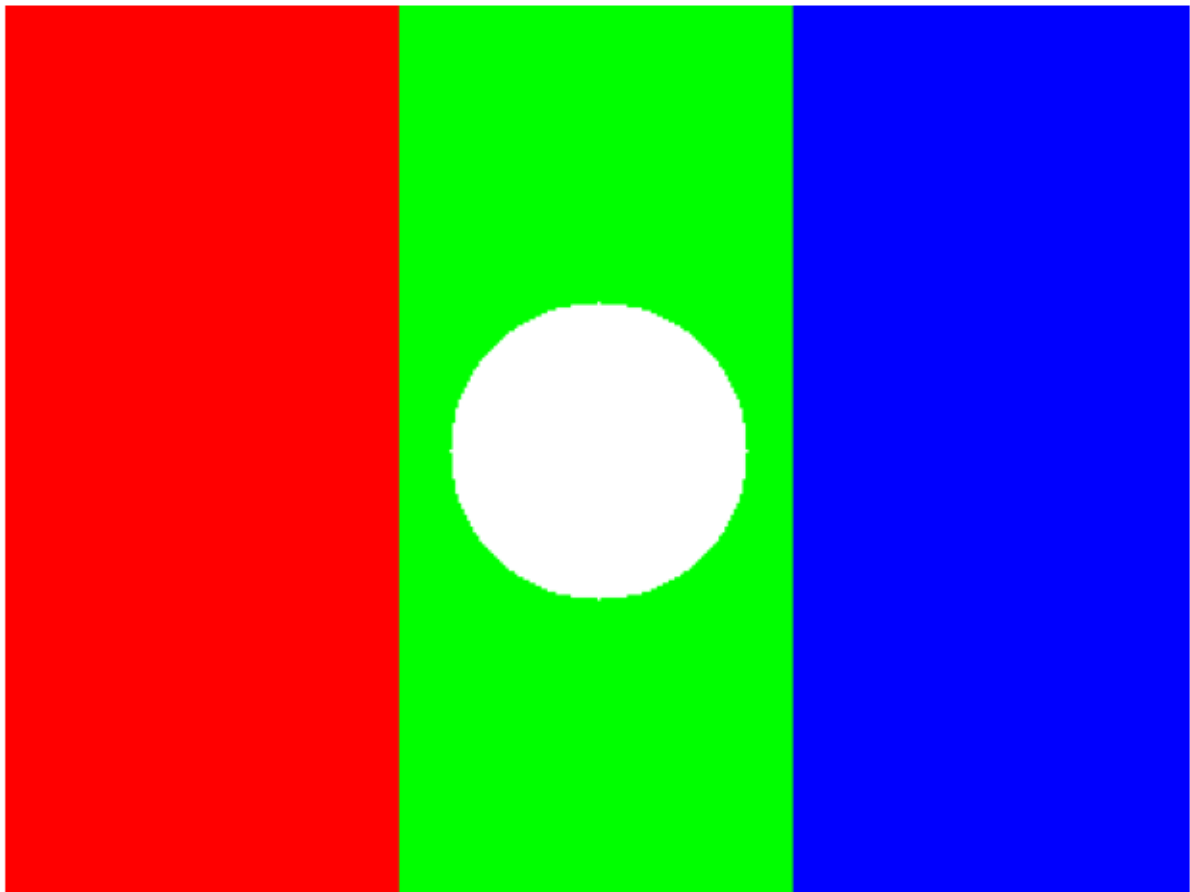



- Terminal/command prompt yang menunjukkan environment aktif
- Output1 dari script test_multimedia.py



- Output2 dari script test_multimedia.py

Test Image with RGB Stripes and White Circle



Gunakan perintah `\includegraphics` untuk menyisipkan gambar

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?
[Menurut saya supaya stabil dan fleksibel untuk eksperimen dan disarankan kawan.]
2. Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?
[Menurut saya uv lebih modern dan cepat dibanding venv, sedangkan conda lebih lengkap karena sudah include package manager. Saya memilih uv karena lebih ringan dan cepat untuk setup, dan juga dibantu kawan dan dianjurkan uv.]
3. Library mana yang paling sulit diinstall dan mengapa?
[Menurut saya tidak ada yang sulit, karena semua lancar.]
4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?
[Menurut saya dengan membuat environment baru atau mengupdate library yang bermasalah.]
5. Jelaskan fungsi dari masing-masing library yang berhasil Anda install!
[Librosa untuk audio processing, soundfile untuk membaca dan menulis file audio, scipy untuk komputasi ilmiah, opencv untuk image processing, pillow untuk manipulasi gambar, scikit-image]

untuk analisis gambar, matplotlib untuk visualisasi data, ffmpeg untuk video processing, moviepy untuk editing video, numpy untuk komputasi numerik, pandas untuk manipulasi data, jupyter untuk notebook interaktif.]

4.4 Troubleshooting

Dokumentasikan masalah yang Anda hadapi (jika ada) dan cara mengatasinya:

- **Masalah 1:** *[Saat mengikuti template latex dan eror]*
Solusi: *[Menggunakan Co Pilot, dan menambahkan usepackage font.]*
- **Masalah 2:** *[Saat mengikuti tutorial output1]*
Solusi: *[Co Pilot, dan langsung dibenarkan.]*
- **Masalah 3:** *[Saat memasukkan gambar dan tidak rapih]*
Solusi: *[Co Pilot dan langsung benar.]*

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

5.1 Untuk Conda

```
1 conda env export > environment.yml
```

Kode 11: Export conda environment

5.2 Untuk venv/uv

```
1 pip freeze > requirements.txt
```

Kode 12: Export pip requirements

Copy-paste isi file environment.yml atau requirements.txt di sini:

```
1 [import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Generate simple sine wave
5 duration = 2 # seconds
6 sample_rate = 44100
7 frequency = 440 # A4 note
8
9 t = np.linspace(0, duration, int(sample_rate * duration))
10 audio_signal = np.sin(2 * np.pi * frequency * t)
11
12 # Plot waveform
13 plt.figure(figsize=(10, 4))
14 plt.plot(t[:1000], audio_signal[:1000]) # Plot first 1000 samples
15 plt.title('Sine Wave (440 Hz)')
16 plt.xlabel('Time (s)')
17 plt.ylabel('Amplitude')
18 plt.grid(True)
19 plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
20 plt.show()
21
22 print(f"Generated {duration}s sine wave at {frequency}Hz")
```

```

23 print(f"Sample rate: {sample_rate}Hz")
24 print(f"Total samples: {len(audio_signal)}")
25 ]

```

Kode 13: Environment/Requirements file

```

1  [import numpy as np
2  import matplotlib.pyplot as plt
3  from PIL import Image
4
5  # Create a simple test image
6  width, height = 400, 300
7  image = np.zeros((height, width, 3), dtype=np.uint8)
8
9  # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 255 # Green section
12 image[:, 2*width//3:, 2] = 255 # Blue section
13
14 # Add a white circle in the center
15 center_x, center_y = width//2, height//2
16 radius = 50
17 Y, X = np.ogrid[:height, :width]
18 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
19 image[mask] = [255, 255, 255]
20
21 # Display and save
22 plt.figure(figsize=(8, 6))
23 plt.imshow(image)
24 plt.title('Test Image with RGB Stripes and White Circle')
25 plt.axis('off')
26 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
27 ]

```

[asd.ipynb](#)

6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia
- Persiapan untuk project multimedia selanjutnya
- Saran untuk mahasiswa lain yang akan melakukan setup serupa

[Kesimpulan saya adalah proses setup ini sangat penting untuk memastikan lingkungan kerja yang stabil dan terorganisir. Dengan menggunakan environment terpisah, kita dapat menghindari konflik antar library dan memastikan bahwa project kita dapat direproduksi dengan mudah. Untuk project multimedia selanjutnya, saya akan lebih memperhatikan dokumentasi dan manajemen dependency. Saran saya untuk mahasiswa lain adalah selalu membaca dokumentasi resmi dari library yang digunakan dan jangan ragu untuk mencari bantuan di komunitas jika mengalami kesulitan.]

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting. [1]

References

- [1] OpenAI, “ChatGPT: Language Model by OpenAI,” <https://openai.com/chatgpt>, 2023, accessed: 2023-10-01.